RadarFX

Users Guide

JRM Technologies
Real Physics • Real Sensors • Real-time

VT MAK
A company of VT Systems

RadarFX

Users Guide

Version 1.4

JRM Technologies
Real Physics • Real Sensors • Real-time

VT MAK
A company of VT Systems

VT MAK
150 Cambridge Park Drive, 3rd Floor
Cambridge, MA 02140 USA

Voice: 617-876-8085
Fax: 617-876-9208

info@mak.com

www.mak.com

Revision RFX-1.4-1-190918

Contents

Chapter 3. Using the RadarFX Simple Client

Chapter 4. Adding GenesisMC Models to RadarFX

Index

Preface

This manual is for persons who will install RadarFX and use it to view simulations. The manual assumes that you are familiar with basic administrative tasks and the graphical window environment for your operating system.

For the latest product information, please see release-specific documentation for your version of RadarFX.

MAK Products

RadarFX is a member of the VT MAK line of software products designed to streamline the process of developing and using networked simulated environments. The VT MAK product line includes the following:

- VR-Link® Network Toolkit. VR-Link is an object-oriented library of C++ functions and definitions that implement the High Level Architecture (HLA) and the Distributed Interactive Simulation (DIS) protocol. VR-Link has built-in support for the RPR FOM and allows you to map to other FOMs. This library minimizes the time and effort required to build and maintain new HLA or DIS-compliant applications, and to integrate such compliance into existing applications.

  VR-Link includes a set of sample debugging applications and their source code. The source code serves as an example of how to use the VR-Link Toolkit to write applications. The executables provide valuable debugging services such as generating a predictable stream of HLA or DIS messages, and displaying the contents of messages transmitted on the network.

- MAK RTI. An RTI (Run-Time Infrastructure) is required to run applications using the High Level Architecture (HLA). The MAK RTI is optimized for high performance. It has an API, RTIspy®, that allows you to extend the RTI using plug-in modules. It also has a graphical user interface (the RTI Assistant) that helps users with configuration tasks and managing federates and federations.
- VR-Forces®. VR-Forces is a computer generated forces application and toolkit. It provides an application with a GUI, that gives you 2D and 3D views of a simulated environment.

  You can create and view entities, aggregate them into hierarchical units, assign tasks, set state parameters, and create plans that have tasks, set statements, and conditional statements. You can simulate using entity-level modeling, which focuses on the actions of individual people and vehicles, and aggregate-level modeling, which focuses on the interaction of large hierarchical units.

  VR-Forces also functions as a plan view display for viewing remote simulation objects taking part in an exercise. Using the toolkit, you can extend the VR-Forces application or create your own application for use with another user interface.

- VR-Vantage™. VR-Vantage is a line of products designed to meet your simulation visualization needs. It includes three end-user applications (VR-Vantage Stealth, VR-Vantage PVD, and VR-Vantage IG) and the VR-Vantage Toolkit.
  - VR-Vantage Stealth displays a realistic, 3D view of your virtual world, a 2D plan view, and an exaggerated reality (XR) view. Together these views provide both situational awareness and the big picture of the simulated world. You can move your viewpoint to any location in the 3D world and can attach it to simulation objects so that it moves as they do.

  - VR-Vantage IG is a configurable desktop image generator (IG) for out the window (OTW) scenes and remote camera views. It has most of the features of the Stealth, but is optimized for its IG function.
  - VR-Vantage PVD provides a 2D plan view display. It gives you the big picture of the simulated world.
  - SensorFX. SensorFX is an enhanced version of VR-Vantage that uses physics based sensors to view terrain and simulation object models that have been materially classified. It is built in partnership with JRM Technologies.
  - The VR-Vantage Toolkit is a 3D visual application development toolkit. Use it to customize or extend MAK's VR-Vantage applications, or to integrate VR-Vantage capabilities into your custom applications. VR-Vantage is built on top of OpenSceneGraph (OSG). The toolkit includes the OSG version used to build VR-Vantage.

- MAK Data Logger. The Data Logger, also called the Logger, can record HLA and DIS exercises and play them back for after-action review. You can play a recorded file at speeds above or below normal and can quickly jump to areas of interest. The Logger has a GUI and a text interface. The Logger API allows you to extend the Logger using plug-in modules or embed the Logger into your own application. The Logger editing features let you merge, trim, and offset Logger recordings.
- VR-Exchange™. VR-Exchange allows simulations that use incompatible communications protocols to interoperate. For example, within the HLA world, using VR-Exchange, federations using the HLA RPR FOM 1.0 can interoperate with simulations using RPR FOM 2.0, or federations using different RTIs can interoperate. VR-Exchange supports HLA, TENA, and DIS translation.
- VR-TheWorld™ Server. VR-TheWorld Server is a simple, yet powerful, web-based streaming terrain server, developed in conjunction with Pelican Mapping. Delivered with a global base map, you can also easily populate it with your own custom source data through a web-based interface. The server can be deployed on private, classified networks to provide streaming terrain data to a variety of simulation and visualization applications behind your firewall.
- DI-Guy™. DI-Guy is an SDK that allows you to embed the DI-Guy library in your real-time application and populate your world with lifelike human characters. DI-Guy provides high-fidelity human characters that move realistically, respond to simple high-level commands, and travel throughout the environment as directed by the hosting visual application. It comes with thousands of ready-to-use characters, appearances, and motions. DI-Guy character simulation is an integral part of all MAK's visual applications, including VR-Engage, VR-Forces, and VR-Vantage.
- RadarFX. RadarFX is a client-server application that simulates synthetic-aperture radar (SAR). The server application, which is based on VR-Vantage and SensorFX, loads a terrain database and, optionally, connects to simulations. A client application requests SAR images from the server. RadarFX includes a sample client application.

- VR-Engage. VR-Engage is a multi-role virtual simulator that lets users play the role of a first person human character, a ground vehicle driver, gunner or commander, a sensor operator, or the pilot of a fixed wing aircraft or helicopter. It incorporates the VR-Force simulation engine and the VR-Vantage graphics rendering capabilities.
- WebLVC Server. WebLVC Server implements the server side of the WebLVC protocol so that web-based simulation federates can participate in a distributed simulation. It is part of the WebLVC Suite, which includes the server and several sample applications that work with VR-Forces and VR-Vantage.

How to Contact Us

For RadarFX technical support, information about upgrades, and information about other MAK products, you can contact us in the following ways:

Telephone

Call or fax us at:	Voice:	617-876-8085 (extension 3 for support)
	Fax:	617-876-9208

E-mail

Sales and upgrade information:	info@mak.com
Technical support:	support@mak.com

Internet

MAK web site home page:	www.mak.com
License key requests:	www.mak.com/support/get-licenses
Product version and platform information:	www.mak.com/support/product-versions
For the free, unlicensed MAK RTI:	www.mak.com/support/bonus-material

Post

Send postal correspondence to:	VT MAK 150 Cambridge Park Drive, 3rd Floor Cambridge, MA, USA 02140

When requesting support, please tell us the product you are using, the version, and the platform on which you are running.

Document Conventions

This manual uses the following typographic conventions:

Monospaced	Indicates commands or values you enter.
Monospaced Bold	Indicates a key on the keyboard.
Monospaced Italic	Indicates command variables that you replace with appropriate values.
Blue text	A hypertext link to another location in this manual or another manual in the documentation set.
{ }	Indicates required arguments.
[ ]	Indicates optional arguments.
|	Separates options in a command where only one option may be chosen at a time.
(|)	In command syntax, indicates equivalent alternatives for a command-line option, for example, (-h | --help).
/	Indicates a directory. Since MAK products run on both Linux and Windows PC platforms, we use the / (slash) for generic discussions of pathnames. If you are running on a PC, substitute a \ (backslash) when you type pathnames.
Italic	Indicates a file name, pathname, or a class name.
sans Serif	Indicates a parameter or argument.
➤	Indicates a one-step procedure.
Menu → Option	Indicates a menu choice. For example, an instruction to select the Save option from the File menu appears as: Choose **File → Save**.
	Click the icon to run a tutorial video in the default browser.
	Indicates supplemental or clarifying information.
	Indicates additional information that you must observe to ensure the success of a procedure or other task.

Directory names are preceded with dot and slash characters that show their position with respect to the RadarFX home directory. For example, the directory *radarfxx.x1.4/doc* appears in the text as *./doc*.

Mouse Button Naming Conventions

An instruction to click the mouse button, refers to clicking the primary mouse button, usually the left button for right-handed mice and the right button for left-handed mice. The context-sensitive menu, also called a popup menu or right-click menu, refers to the menu displayed when you click the secondary mouse button, usually the right button on right-handed mice and the left button on left-handed mice.

Third Party Licenses

MAK software products may use code from third parties. This section contains summary license information and where required, specific license documentation required by these third parties.

The following libraries are covered by the GNU Lesser General Public License (LGPL) license:

- CIGI
- DevIL
- GEOS
- GStreamer
- LibIconV
- OPCODE
- OpenAL
- OSG
- OsgEarth
- Pthread
- Qt
- Qwt
- LibWebSockets

The following libraries are covered by the ZLib license:

- Bullet Physics
- Zlib
- LibPNG
- LibSDL
- Minizip

The following libraries are covered by the MIT license:

- Expat
- FreeGLUT
- GDAL
- GeoTiff
- HarfBuzz
- LibCURL
- LibSquish
- LibXML
- LUA
- LuaBind
- Proj

The following libraries are covered by the BSD license:

- FreeType
- GLEW
- JPEG
- LibTiff
- Protobuf
- PCRE

The following libraries are covered by the commercial licenses:

- SpeedTree
- Sundog Triton
- Sundog SilverLining
- FlexLM
- Gameware
- JRM Technologies SenSimRT and SigSimRT
- GLStudio
- OpenFlight
- CDB API

The following libraries are covered by custom licenses:

- NVidia CG Toolkit
- FreeImage (FreeImage Public License)
- Boost (Boost Software License)
- Poco (Boost Software License)

COLLADA DOM is covered by the SCEA Shared Source license.

Thread Building Blocks is covered by the GPL license:

SQLLite is in the public domain.

Boost License

VR-Link, and all MAK software that uses VR-Link uses some code that is distributed under the Boost License. All header files that contain Boost code are properly attributed. The Boost web site is: www.boost.org.

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

libXML and libICONV

VR-Link and all MAK software that uses VR-Link, links in libXML and libICONV. On some platforms the compiled libraries and header files are distributed with MAK Products. MAK has made no modifications to these libraries. For more information about these libraries please see the following web sites:

- The LGPL license is available at: http://www.gnu.org/licenses/lgpl.html.
- Information about IconV is at: http://www.gnu.org/software/libiconv/.
- Information about LibXML is at: http://xmlsoft.org/.

1. Introduction to RadarFX

This chapter provides a brief introduction to RadarFX and explains how to install it. You must also install the MAK license manager. To use RadarFX with HLA, you must install an RTI.

1.1. Introduction to RadarFX

RadarFX (RadarFX) is a client-server application that can simulate synthetic-aperture radar (SAR) (Spot SAR and Strip SAR) and inverse synthetic aperture radar (ISAR). The server application, which is based on VR-Vantage and SensorFX, loads a terrain database and, optionally, connects to simulations. A client application requests images from the server. RadarFX includes a sample client application.

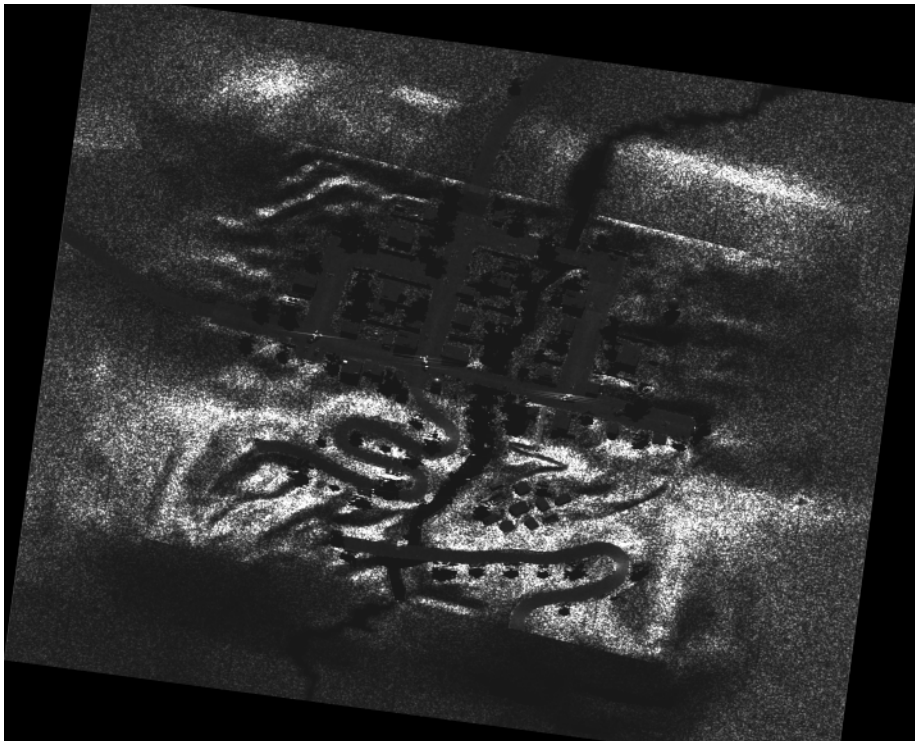

Figure 1-1. RadarFX SAR image

RadarFX includes the documentation for VR-Vantage and SensorFX, which completely explain how to use those products. This manual provides only enough information about those applications to allow you to use RadarFX.

Client applications can be standalone, like the simple client included with RadarFX or they can be embedded in other applications. For example, if you are controlling an aircraft from VR-Engage, you can request images from the SAR multifunction display (Figure 1-2).

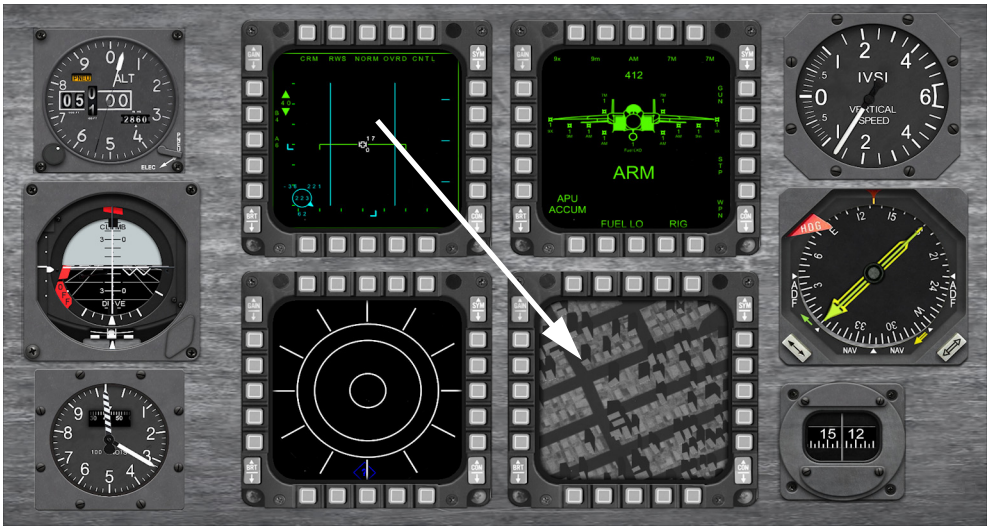

Figure 1-2. SAR image

1.2. Installing RadarFX

A full RadarFX installation includes the following components:

- The application.
- The data package, which consists of five separate files.
- The License Manager software.
- For HLA operation, an RTI (or Runtime Infrastructure.)

Before you install RadarFX, please read the Release Notes to see if there are any special instructions for installation.

Note the following:

- When you install large applications on Windows, 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.

RadarFX is provided as executable installer files on DVD, or as downloaded files. The installers are named to indicate the compiler used to build that version of RadarFX.

To install RadarFX:

1. Make sure that the application installer and the data package files are in the same directory.
2. Run the application installer.
3. When you are prompted to choose an option for installing the data package (Figure 1-3), select the default - “Install the recommended data package automatically”.

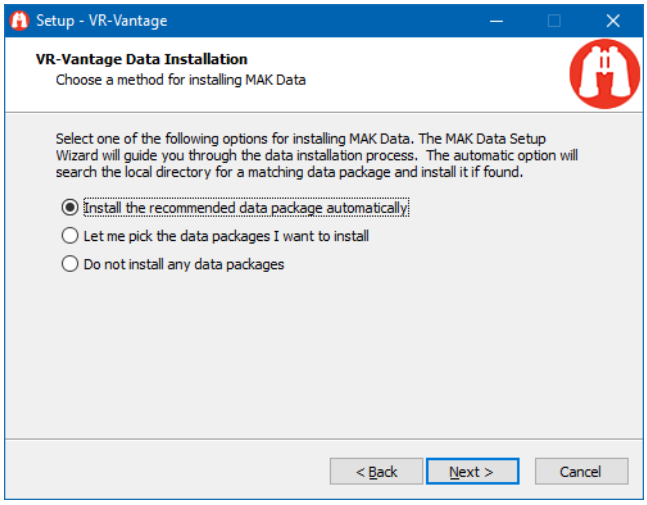

Figure 1-3. Data installer selection page

4. Continue with the installation wizard. When the application installer finishes, the MAK Data Setup Wizard (similar to Figure 1-4) opens automatically and installs the data.

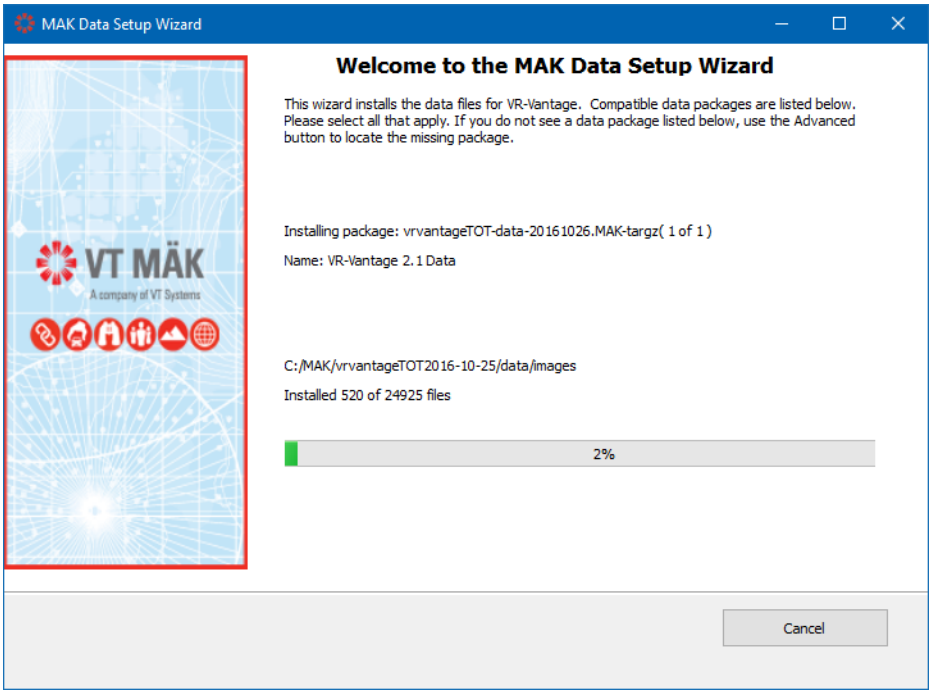

Figure 1-4. MAK Data Setup Wizard

! You must install the data package. It is not optional. RadarFX will not run properly without it. The data package consists of three generically named files, one terrain file, and one file containing sensorFX data.

1.2.1. Uninstalling RadarFX

The RadarFX uninstaller only uninstalls the application. It does not uninstall the data package.

To uninstall RadarFX:

1. In the RadarFX root directory, run *unins000.exe*.
2. Delete the RadarFX root directory.

1.3. The RadarFX Directory Structure

The installation process creates the RadarFX directory tree shown in Figure 1-5 (runtime and toolkit).

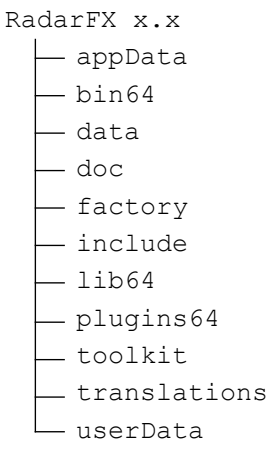

Figure 1-5. RadarFX directory structure

Table 1-1 describes the contents of the directories.

Table 1-1: Contents of RadarFX directories

Directory	Directory contents
./appData	Subdirectories contain data required by the RadarFX applications, such as model definitions and saved.
./bin64	Executables and dynamic link libraries.
./data	Sample databases, images, and vehicle models.
./doc	RadarFX documentation and class reference files.
./factory	Default settings (for reverting to "factory").
./include	Header files for the RadarFX Toolkit API.
./lib64	Libraries for building applications with the RadarFX Toolkit API.
./plugins64	Plug-in shared libraries.
./toolkit	Examples for the RadarFX Toolkit API.
./translations	Translation files for localizing RadarFX menus.
./userData	Terrain and scene files saved by application users.

1.4. Installing and Setting Up the MAK License Manager

Before you can use a MAK product, you must obtain a valid license file, install the MAK License Manager, and configure the license server and client machines. The License Manager uses a client-server architecture, so you do not need to install the License Manager on every computer on which you install MAK products. You only need to install it on the computer that you will use as the license server.

If you have already installed the License Manager for another MAK product, you do not have to install it again. You just need to make sure you have licenses for your newly installed products.

The License Manager installer is included on MAK installation media. It is separate from the product installers. You can download the installers from our web site at https://www.mak.com/support/license-support.

Complete installation and configuration instructions are included with the License Manager installer. Instructions are also available at the license support page.

Some customers use dongle licenses instead of running a license server. Instructions for using dongles are in the License Manager documentation.

1.4.1. Specifying the License Server

The first time you run a MAK application on a particular computer, the License Setup dialog box opens (Figure 1-6). It prompts you to enter the hostname of the license server and optionally, a port number.

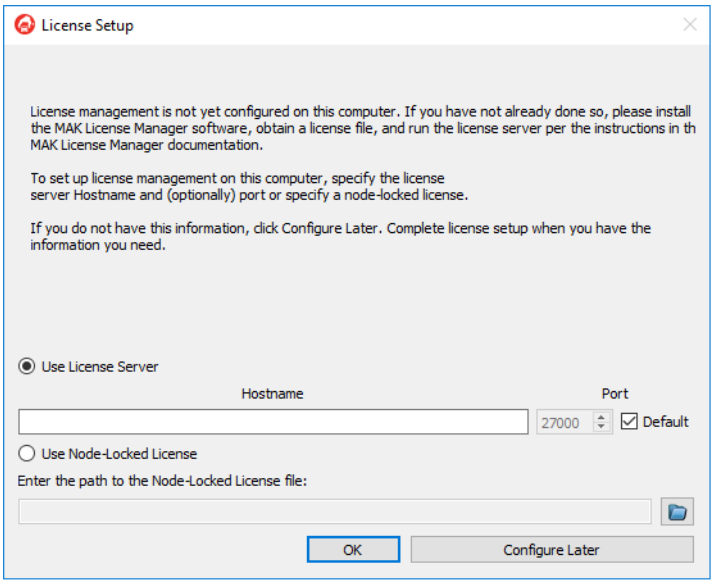

Figure 1-6. License Setup dialog box

If you do not know the hostname of the license server, click Configure Later. When you have the hostname, you can start the application again and complete the dialog box. You will not be able to run any MAK applications until you set up license management.

If you know the hostname, type it in the Hostname box and then click OK. The application will start.

Under limited circumstances, MAK issues node-locked licenses. If you have a node-locked license, select the Use Node Locked License option and enter the path to the license file.

- If you are running MAK products on the license server machine, it is also a client, so you must specify the license server on that machine, too.
- If you change the license server, the saved configuration will no longer be valid and the License Setup dialog box will open the next time you start a MAK application.
- You can clear the saved license configuration by deleting the cache file. On Windows, it is *C:\Users\user_name\AppData\Roaming\MAK\licenses<n>.txt*. (The *AppData* directory is hidden by default.) On Linux, it is *.mak/license<n>.txt*. (There may be more than one cache file, for example, *licenses1.txt* and *licenses2.txt*.)

The MAKLMGRD_LICENSE_FILE Environment Variable

As an alternative to using the License Setup procedure described in the previous section, you can configure the license server in an environment variable. The MAKLMGRD_LICENSE_FILE environment variable identifies the server machine. If you set this environment variable, it overrides the settings stored by the License Setup procedure.

The syntax for the environment variable is: `@Server_name`. For example, if the server machine is oak, set the environment variable to `@oak`.

The following sections explain how to set environment variables on the different platforms that MAK products run on.

Windows

To add the MAKLMGRD_LICENSE_FILE in Windows:

1. On the Start menu, click the Settings button. The Settings window opens.
2. In the search box, type environment. A list of choices is displayed.
3. Select Edit the System Environment Variables. The System Properties dialog box opens.
4. Click Environment Variables. The Environment Variables dialog box opens.
5. In the System Variables group box, click New. The New System Variable dialog box opens.
6. In the Variable Name field, enter MAKLMGRD_LICENSE_FILE.
7. In the Variable Value field, enter `@server_name`, where `server_name` is the name of the license server.
8. Click OK to back out of each dialog box and set the variable.

Linux

On Linux, you set environment variables in your *.cshrc* (or equivalent startup file). Set the variable similarly to the following example:

```
setenv MAKLMGRD_LICENSE_FILE @oak
```

If you are using the sh or bash shells, you set environment variables in your *.profile* file (or *.bashrc*). Set the variable similarly to the following example:

```
MAKLMGRD_LICENSE_FILE=@oak
export MAKLMGRD_LICENSE_FILE
```

Do not put spaces around the equal (=) sign.

You are ready to run the license server and use your new licenses or MAK products.

1.5. Installing an RTI

An RTI is a software library (and perhaps supporting executables) that implements an HLA Interface Specification. In HLA, applications exchange FOM data through RTI calls, which means that all HLA applications need to use an RTI.

Because of differences in the low-level network mechanisms used by different RTI implementations (which include, but are not limited to packet layout), applications that want to interoperate in the same federation execution must use the same RTI implementation.

Because RTIs are usually provided as dynamic libraries that implement a fixed API, a federation can often switch from one RTI implementation to another between runs (without even recompiling the applications), but during each run, all participants must agree on which RTI to use, much as they must also agree on which FOM to use.

For the most recent information about the RTI versions supported by MAK products, please see the release notes for your MAK application.

1.5.1. Installing the MAK RTI

To install the MAK RTI, follow the instructions in Chapter 2 of *MAK RTI Users Guide*.

Configuring Your System to Use the MAK RTI

The RTI dynamic libraries must be located somewhere on your dynamic library search path. The path is specified in the PATH environment variable.The path is indicated by an environment variable as follows:

- Linux — LD_LIBRARY_PATH.
- Windows — PATH.

The MAK RTI needs to know where to find the following configuration files:

- The federation configuration file (*.fed*, *.fdd*, or *.xml*) for your federation execution (required).
- The RID file (*rid.mtl*) (optional).

Put the configuration files in the directory from which you are running, or set the environment variable RTI_CONFIG to the directory that contains them.

Running Applications with the MAK RTI

To run a MAK application with the MAK RTI:

1. Be sure the license server is running.
2. Start the application. The RTI Assistant will prompt you to choose an RTI configuration.
3. Choose a configuration. If necessary, start the rtiexec.
4. Click Connect. The application should run.

In many cases, you do not need to run the rtiexec to use the MAK RTI, however you can run the rtiexec if you want to. (It is required to use certain features of the MAK RTI.) For more information, please see your RTI documentation.

1.6. Localizing the Graphical User Interface

You can localize your MAK product's graphical user interface so that it uses languages other than English. Your product includes a file called *./translations/product_untranslated.ts*, for example *logger_untranslated.ts* or *vrforces_untranslated.ts*. The file contains entries for all of the application-specific menu and dialog box text. The Qt Linguist tool, which is included with RadarFX, lets you map the English text provided by MAK to your chosen language.

To edit a resource file:

1. Start the Qt Linguist utility. It is *./translations/linguist.exe*
2. In the Qt Linguist window, choose **File** → **Open**. Select the file *./translations/product_untranslated.ts*. The Settings for filename dialog box opens (Figure 1-7).

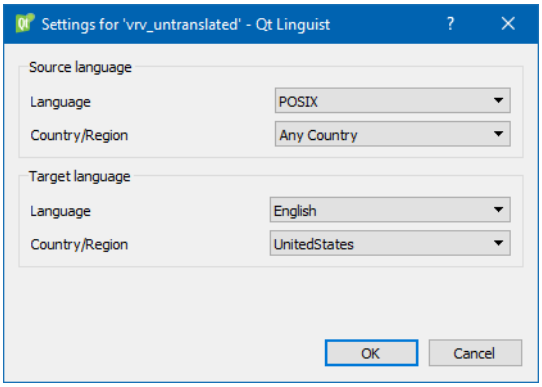

Figure 1-7. Linguist settings dialog box

3. Select values for the source language, target language, and target region or country.
4. Click OK. The translation file is opened and a list of classes is displayed on the left side of the window (Figure 1-8).

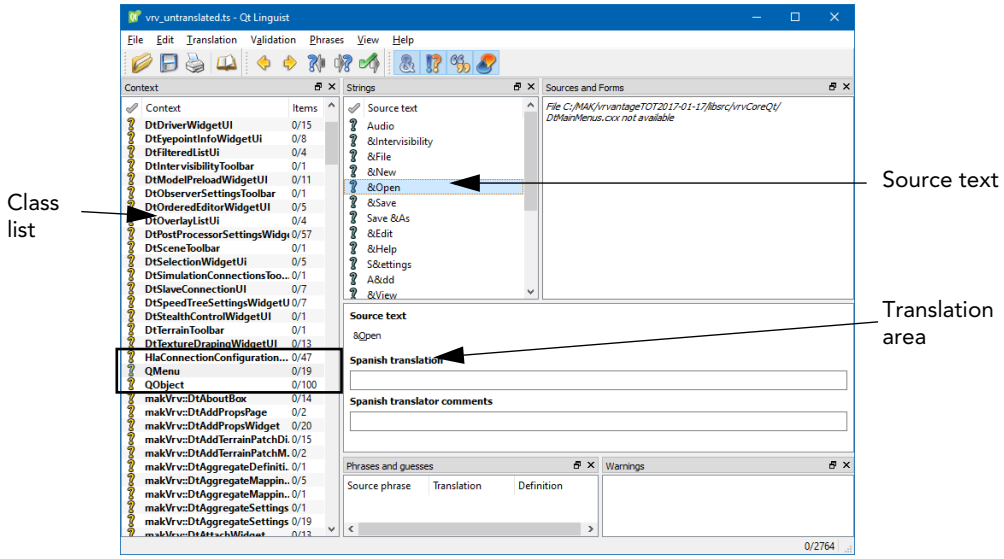

Figure 1-8. Linguist window

5. To translate text, you select each class in turn and translate the text associated with that class. You do not have to understand anything about what the classes mean, you just have to translate the text. In Figure 1-8, *QMenu* is selected. The Strings window lists text that can be translated. Text that has not been translated has a question mark in the checkmark column.

6. In the Source Text list, select the word you want to translate. The text is displayed under Source Text in the translation area (the window below the source text list). Figure 1-9 shows the text Open selected.

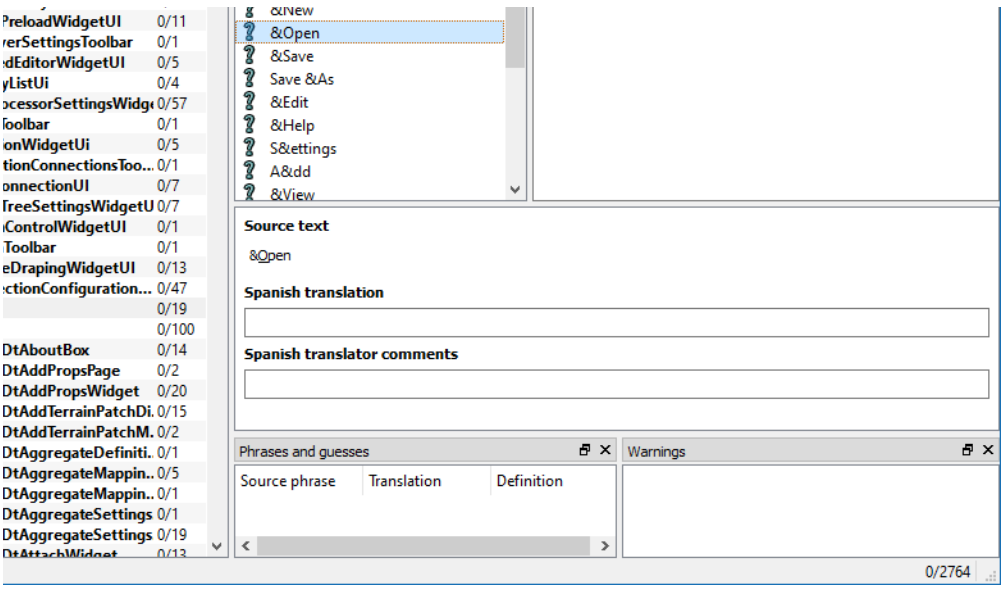

Figure 1-9. Text selected for translation

7. Place the cursor in the text box below *Language* Translation, where *Language* is the target language that you selected when you opened the translation file.
8. Type the translation text.
9. When you are done and you are certain of the translation, select the Done and Next button ( ) on the toolbar. A green check mark replaces the question mark (Figure 1-10). If you move to a different text string without clicking Done and Next, a yellow question mark is displayed for the string you translated. When you translate all elements for a class, a green check mark replaces the question mark next to the class name.

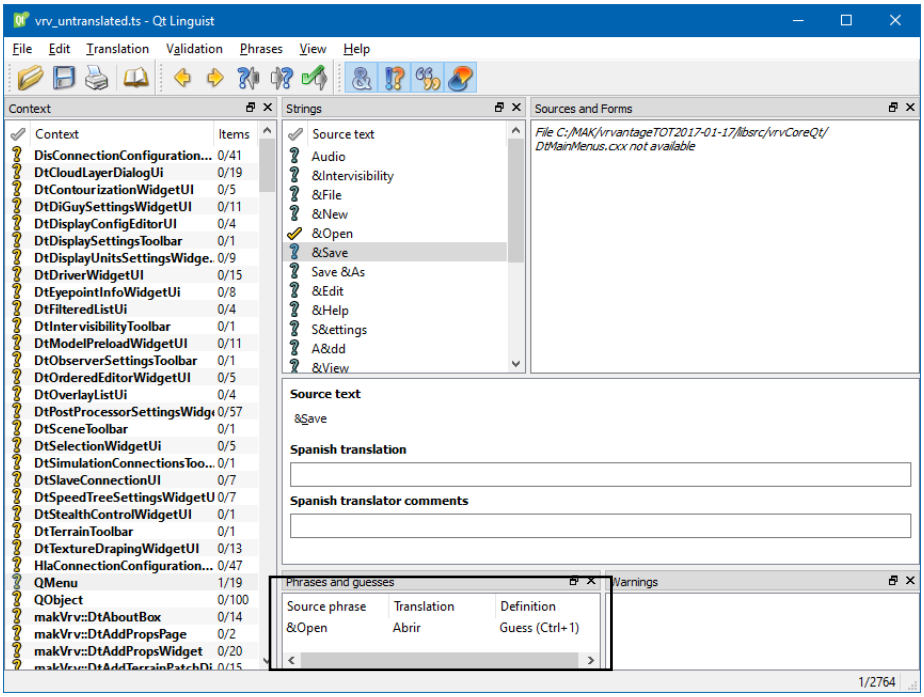

Figure 1-10. Completed translation for Open

10. Repeat this process for each element of source text for each class.
11. Choose **File** → **Release**. The file is saved in the format *filename.qm*.
12. Rename the file to a name that indicates the language to which you have translated the file, using the two-character abbreviation typically used with your language. For example, a Spanish translation would be renamed *product_es.qm*.
13. Save your translated *.ts* file.

1.6.1. Translating Other Interface Files

The *radarfx_untranslated.ts* file contains all the text used in the RadarFX interface. However, there may be other text and messages that are not specific to RadarFX that are generated by the software that implements the RadarFX GUI. To translate this text, you need to translate the file *qt_untranslated.ts*. Follow the same procedure as you did for *radarfx_untranslated.ts*.

1.6.2. Applying the Language Files

To put the language file into effect, use one of the following methods:

- Set your computer's environment to the proper language (LANG environment variable).
- Set the locale to the proper language.
- Start RadarFX with the `-G` *`locale`* command line option, where *`locale`* is the two-letter country code for the local language.

2. Using RadarFX Server

This chapter explains how to start and configure RadarFX.

2.1. Starting RadarFX Server

To use RadarFX, you run the server and a client application. The client could be embedded in any third party application.

To start RadarFX:

1. On the Start menu, choose **MAK Technologies** → **RadarFX 1.2.1** → **RadarFX Server**.

---

i The Start menu for Windows 10 does not support submenus. The path would be **MAK Technologies** → **RadarFX Server**. This difference applies to all Start menu paths described in this manual.

RadarFX Server starts and the Load Terrain dialog box opens.

! The VC 12 version of RadarFX uses a batch file to start. If you decide to start it from the command line or a file explorer, you must run *radarFX.bat*, not *radarFX.exe*.

---

2. Select a terrain to load. RadarFX Server loads the terrain and displays it in the window (Figure 2-1).

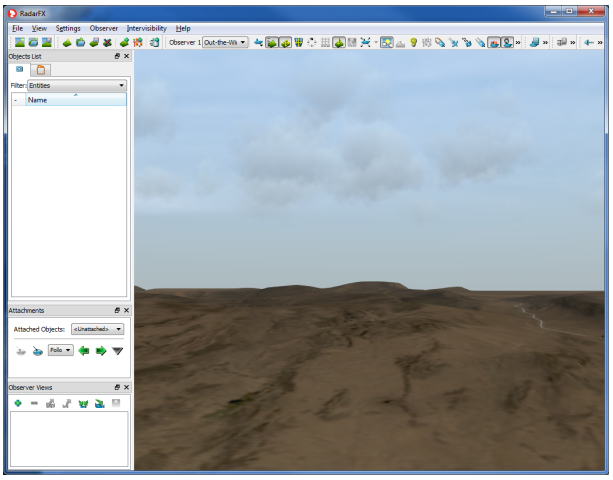

Figure 2-1. RadarFX Server

- Since RadarFX uses SensorFX, you will get best results with a terrain that has been materially classified for its use, or which supports material classification on demand. For details about material classification, please see *SensorFX Users Guide*. A materially classified terrain, *VR-VillageSensorFX.mtf* is included with RadarFX.
- RadarFX does not support geocentric terrains if they are part of a terrain that does not include an osgEarth terrain patch. For more information, please see "Using RadarFX with Geocentric Terrains," on page 2-9.

2.2. Opening a Terrain

The previous section explained how to load a terrain from the Load Terrain dialog box when RadarFX starts up. However, if you choose not to load a terrain immediately, or want to load a different terrain at any time after startup, you can do so. For details, please see Section 3.3, "Opening a Terrain," in *VR-Vantage Users Guide*.

2.3. Connecting to a Simulation

You do not need to connect to a simulation to use a RadarFX client with the RadarFX server. However, if you do not connect to a simulation, you will not be able to view any entities in the server or use them as the viewpoint from which to capture an image.

To connect to a simulation, follow the instructions in Section 3.4, "Connecting to a Simulation Network," in *VR-Vantage Users Guide*.

2.4. Configuring RadarFX Server

You can configure the following settings for RadarFX Server:

- Server Listen Port. The port that the server is listening on for incoming connections from clients. Changing this value restarts the server immediately.
- Default Observer Mode Name. The default observer mode to use for image captures.
- Default Sensor Mode Name. The default sensor mode for image captures.
- Paging Terrain Timeout. If the terrain does not finish paging in by the time the timeout value has been reached, take the image.
- NITF Time Stamp Source. The source of the time stamp to include with an image as metadata. Metadata is published only when sending National Imagery Transmission Format (NITF) images. (For details about choosing the image format, please see "Configuring Image Properties," on page 3-11.)
- Flip Images Vertically, Flip Images Horizontally. Flip the image in the indicated direction before it is sent.

All of the settings except the server listen port are configured in the Application Settings dialog box. The server listen port is configured in the Connection Settings dialog box. RadarFX Server settings are saved automatically when the application exits.

To configure RadarFX Server application settings:

1. Choose **Settings** → **Application**. The Application Settings dialog box opens.
2. Select the RadarFX Server Settings page (Figure 2-2).

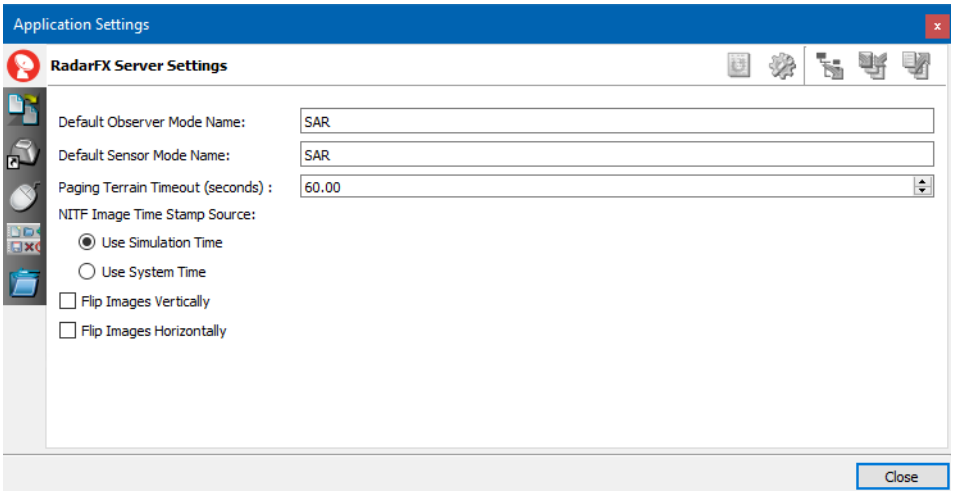

Figure 2-2. RadarFX Server Settings page

3. Change parameter values as desired.

2.4.1. Setting the Server Listen Port

The server listen port is the port that the server is listening on for incoming connections from clients. Changing this value restarts the server immediately.

To set the server listen port:

1. Choose **Settings** → **Connection**. The Simulation Connections dialog box opens.
2. Select the RadarFX Client Connection page (Figure 2-3).

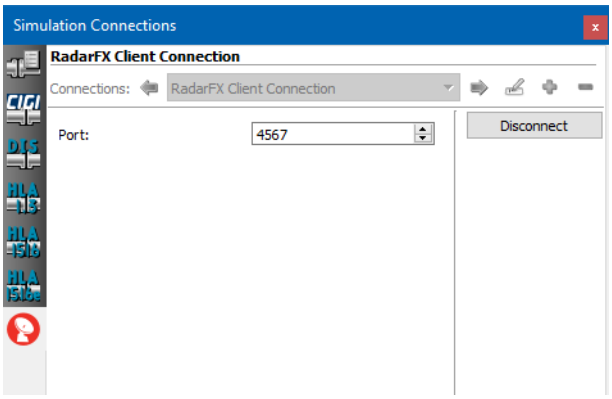

Figure 2-3. RadarFX Client Connections page

3. In the Port box, specify the port you want to use.
4. Click Close.

2.5. Configuring Sensors

You configure sensors on the Sensor Settings page of the Display Settings dialog box. For general details about configuring sensors, please see Section 1.5, "Configuring SensorFX Sensors," in *SensorFX Users Guide*.

The options on the Effects tab are the same for SAR and MMW as they are for LWIR. They are described in Section 1.5, "Configuring SensorFX Sensors," in *SensorFX Users Guide*.

The options on the Advanced tab are the same for all sensor modes. They are described in Section 1.5, "Configuring SensorFX Sensors," in *SensorFX Users Guide*.

The ISAR sensor mode has one option on the Effects tab. You can enable or disable ISAR post processing.

The parameters for the SAR, ISAR, and MMW sensors are described in Table 2-1.

RadarFX Server uses SensorFX sensors. The discussion of CameraFX sensors in *VR-Vantage Users Guide* does not apply.

Table 2-1: Sensor parameters for SAR, ISAR, and MMW sensors

Parameter	Description
AGC	Automatic Gain Control. With AGC on, the maximum received power (and thus saturation ratio) will be computed as the power estimated to be returned by a surface of unit less areal RCS = 0.25 at the boresight range, disregarding atmospherics, but including system noise. Default: OFF.
Doppler Coloring	(MMW only) A display parameter that color-codes returns according to their perceived velocity relative to the sensor. The intensity of the return is not altered, but the color shifts to red (away from sensor) or blue (toward sensor) by an amount proportional to the position of the return in the detectable Doppler scale, the maximum velocity of which is governed by the wavelength and pulse repetition frequency (PRF), and the minimum of which is determined by the wavelength and time-on-target. ON or OFF. Default: OFF
Toggle Image Orientation	(SAR only) Rotates the image orientation by 180 degrees.
Region of Interest Down-range	(SAR only) The desired downrange extent of the section of terrain to be mapped, relative to the current boresight terrain intersection point. Nominal values in range 100-1000.
Region of Interest Cross-range	(SAR only) The desired cross range extent, in meters, of the section of terrain to be mapped, relative to the current boresight terrain intersection point. Nominal values in range 100-1000.
Doppler Spatial Effects	(SAR only) Enables the Doppler spatial effects in the SAR sensor mode. These effects show up on moving entities as either a blurring or a displacement of the entity. When enabled, these effects have a significant impact on the performance of the rendering.

Table 2-1: Sensor parameters for SAR, ISAR, and MMW sensors

Parameter	Description
Center Frequency	The carrier frequency of the radar. It affects atmospheric transmission loss and noise, beam width, Doppler resolution, and wavelength. Higher values improve crossrange positional resolution (beam width), and downrange Doppler resolution, at the expense of a smaller effective receiver area. Typical values are: ◆ L-band: 1-2 GHz ◆ S-band: 2-4 GHz ◆ C-band: 4-8 GHz ◆ X-band: 8-12 GHz ◆ Ku-band: 12-18 GHz ◆ Ka-band: 30-40 GHz ◆ MMW band ~ 40 - 300 GHz ◆ W-band: 90-100 GHz.
Pulse Width	The width of the radar transmitter's pulses, in microseconds. It is inversely proportional to bandwidth. Lower values improve downrange positional resolution and increase the received signal. Higher values reduce system noise. Should be smaller than 1/PRF. Typically 1.E-9 to 1.E-2 [s].
Pulse Repetition Frequency	The frequency, in Hz, with which the transmitter sends pulses of RF power. Lower values increase the maximum unambiguous range and received signal, but decrease maximum detectable Doppler shift. Should be smaller than 1/PW and c/2DR, where c is speed of light and DR is downrange region of interest. Should be greater than about V/L, where V is intended aircraft speed and L is integration path length. Typically 250-300000 Hz.
Transmitter Polarization Angle	Linear polarization angle of transmitter gain, in degrees. Typical values: ◆ Horizontal = 0 ◆ Vertical = 90. Default = 45.
Transmitter Directivity	Similar to a lobe-width parameter, in that larger values focus the transmitter gain pattern into a smaller, but stronger, main lobe centered at the boresight direction. A value of 1 results in no focusing, that is, a Unity gain pattern. Must be greater than zero, but typically set > = 1.

Table 2-1: Sensor parameters for SAR, ISAR, and MMW sensors

Parameter	Description
Receiver Polarization Angle	Linear polarization angle of receiver acceptance pattern, in degrees. Typical values: ◆ Horizontal = 0 ◆ Vertical = 90. Default = 45.
Receiver Directivity	Similar to a lobe-width parameter, in that larger values focus the receiver acceptance pattern into a smaller but stronger main lobe centered at the boresight direction. A value of 1 results in no focusing. Increasing values also increase effective receiver area. Must be greater than zero, but typically set > = 1.
Integration Path Length	(SAR only) The distance over which the radar collects returned pulses, also known as the SAR antenna length. Higher values improve cross range resolution. Should be greater than desired cross range region of interest, and greater than c/(2ðf), where c is light speed and f is center frequency in Hz. The effect of very high values will be limited by terrain texture resolution, display resolution, or both. Typically 100-10000
Integration Time	(ISAR only) The amount of time spent integrating the radar returns before generating the final ISAR result. Must be greater than zero.
Emitter Power	The power emitted by the radar transmitter. Higher values improve effective image gain. Decreasing power results in a darker image. Must be greater than zero. Typically 50-5000.
System Temperature	Unavoidable internal radar system noise temperature. Higher values degrade signal-to-noise ratio. Must be greater than zero. Typically 270-850 degK.
Saturation S/N Ration	The Signal-to-Noise ratio at which the return pulses saturate the receiver electronics. In other words, this ratio defines the maximum received power of the system, when multiplied by the noise power. Higher values will decrease image intensity. Must be greater than zero, but typically set > = 1.
Gain Pattern	Allows focusing of the emitted or received RF power into lobes. Both the transmitter and receiver share the same gain/acceptance pattern, but can have different directivities. Possible values: ◆ Unity (no lobes). ◆ Lobe (single-lobe). ◆ Rect. Rectangular (main lobe plus side lobes).

Table 2-1: Sensor parameters for SAR, ISAR, and MMW sensors

Parameter	Description
Display Type	Specifies which computed ratio the display is sensitive to. ◆ Scaled Power: Ratio of received power to maximum power. ◆ Scaled Radiance: Ratio of received radiance to maximum radiance. ◆ Scaled Effective RCS: Ratio of received power to power estimated to be returned by a surface of unit RCS at the boresight range, disregarding atmospherics, but including system noise.
Intensity Modulation Gain	Together with the geometry-model-specific material classification (IMOD) files, this governs the extent to which detail variance present in the RGB texture for the entity or terrain models are translated into additional RCS variance in the radar scene.

2.6. Using RadarFX with Geocentric Terrains

If you load a geocentric terrain without using an earth file, RadarFX does not work properly. You can work around this problem by creating an earth file that does not do anything and creating an MTF file that loads the earth file and the terrain.

The earth file should look like this:

```
-->
<map name="MAK Earth" type="geocentric" version="2">
<options>
<terrain color = "#00000000" />
</options>
</map>
```

1. In RadarFX Server, load the earth file as a terrain server.
2. Load the geocentric terrain as a terrain patch.
3. Save the terrain as an MTF terrain.

By default, the observer will be in space, so create a saved view of the terrain at a useful location.

3. Using the RadarFX Simple Client

This chapter explains how to use the RadarFX Simple Client.

3.1. Starting the Simple Client

The typical use of RadarFX is to embed its functionality into a third-party client application. The client application requests an image from RadarFX Server, which provides an image based on the location information sent from the client. The RadarFX Simple Client demonstrates what you can do with the RadarFX API.

To start the RadarFX Simple Client:

1. On the Start menu, choose **MAK Technologies** → **RadarFX Simple Client**.

   RadarFX Simple Client opens and the Connect to RadarFX Server dialog box opens.

2. In the Connect to RadarFX Server dialog box, accept the default values. Click OK. The RadarFX Simple Client opens (Figure 3-1).

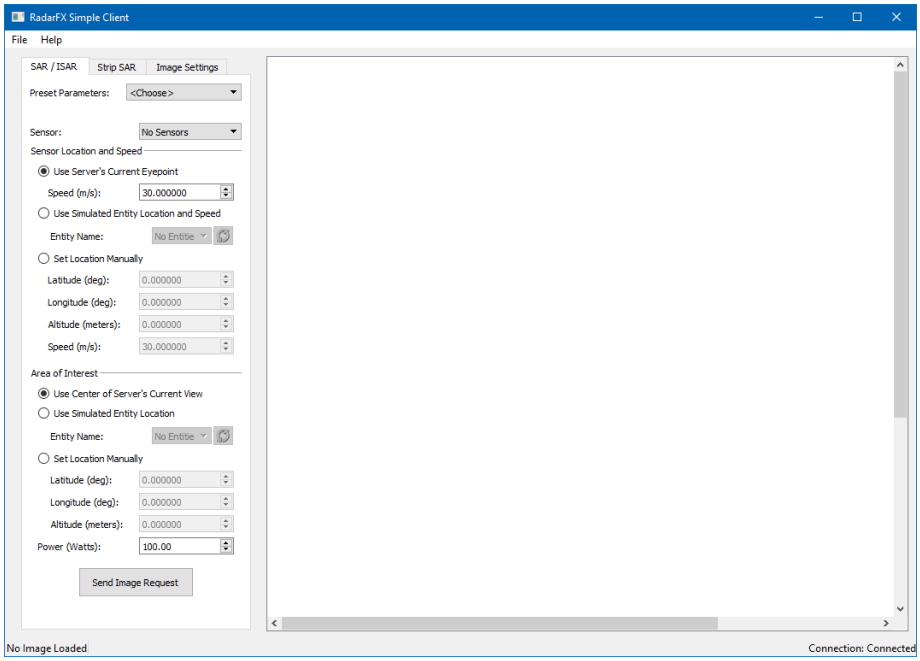

Figure 3-1. RadarFX Simple Client

3.2. Requesting an Image

SAR and ISAR images are requested the same way. Strip SAR images are requested differently. A SAR image is returned as one image. ISAR can return multiple images. Strip SAR returns a composite image composed of the number of strips that you specify.

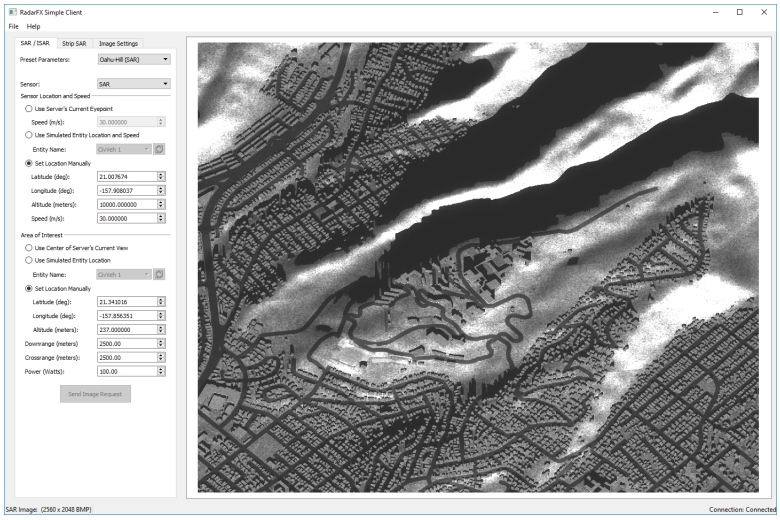

Figure 3-2. SAR image

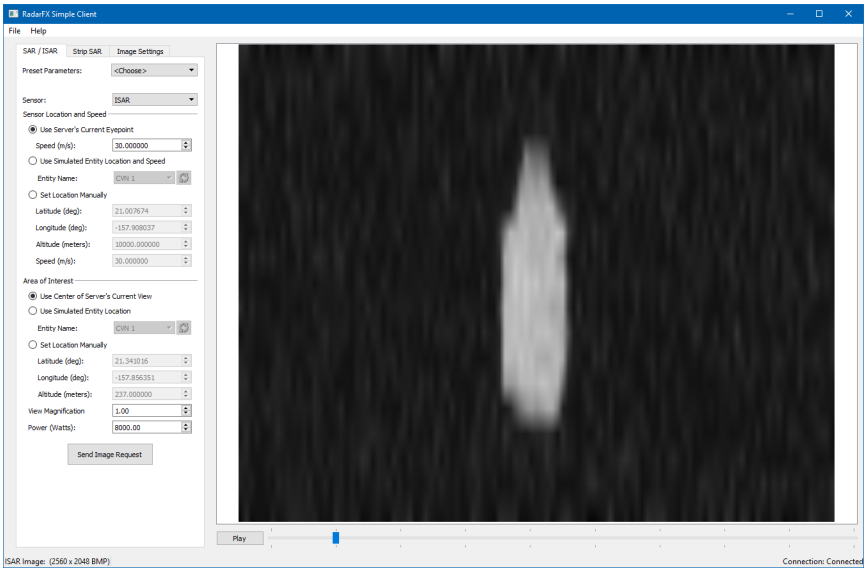

Figure 3-3. ISAR image

3.2.1. Requesting a SAR or ISAR Image

You can create presets for frequently used SAR and ISAR configuration settings so that you do not have to enter them every time you use the client application. For details, please see "Creating a Configuration Preset," on page 3-12.

When streaming terrain from a remote server such as VR-TheWorld Server, it is possible that the first time you request a SAR image for a particular area, the SAR image will be generated before all of the terrain pages are finished streaming in. This is less of an issue once you have cached the data around your area of interest to your local hard drive, but still can be slower than expected. The workaround is to make sure your target area is fully paged in before requesting SAR images.

To request a SAR or ISAR image from the server:

1. Select the SAR/ISAR tab.
2. Specify the sensor to use. For details, please see "Specifying the Sensor," on page 3-4.
3. Specify the location of the sensor. For details, please see "Specifying the Location of the Sensor," on page 3-5.
4. Specify the area of interest. For details, please see "Specifying the Area of Interest," on page 3-6.
5. For SAR, specify the downrange and crossrange values. For details, please see "Specifying the Area of Interest," on page 3-6.

   For ISAR, specify the view magnification.
6. Set the power to use. For details, please see "Specifying the Power," on page 3-7.
7. Optionally, change Image settings. For details, please see "Configuring Image Properties," on page 3-11.
8. Send the request. For details, please see "Requesting the Image," on page 3-7.

Specifying the Sensor

The client gets a list of sensors from the server.

To specify the sensor to use:

1. In the RadarFX Simple Client window, select the SAR/ISAR tab.
2. Select an option in the Sensor list.

Specifying the Location of the Sensor

You must tell the server where the sensor that will take the image is located. You can:

- Use the location of the server's observer. This option is most useful when both the server and the client are running on the same machine and you can manually move the observer into position and then request a radar image. The eyepoint should be above the target location.
- Attach to an entity in the simulation (if the server is connected). The client gets a list of entities from the server. The client does not need to be connected to the simulation. The eyepoint should be above the target entity.
- Enter a location manually in latitude, longitude, and altitude.

To specify a location:

1. In the Sensor Location and Speed section, choose one of the options.
2. If you choose Use Simulated Entity Location and Speed, select an entity in the Entity Name list.

   If you choose Set Location Manually, specify Latitude, Longitude, Altitude, and Speed.

Specifying the Area of Interest

The area of interest is the area that you want to take an image of. The sensor does not have to be located in the area, although it could be. You can specify the area of interest as:

- An area centered on the server's eyepoint.
- An area centered on a simulated entity.
- An area specified by its latitude, longitude, and altitude.

The cross range and down range values specify the size, in meters, of the area of interest centered on the target location, as illustrated in Figure 3-4.

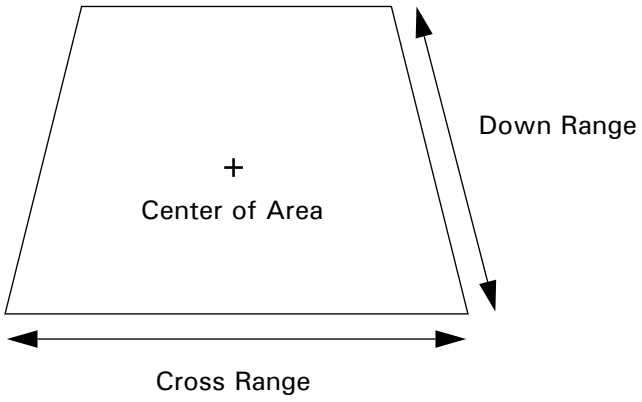

Figure 3-4. Area of interest

ISAR works best if you specify an entity as the target. The entity must be a materially classified ship.

To specify the area of interest:

1. In the Area of Interest section, select an option.
2. If you choose Use Simulated Entity Location, select an entity in the Entity Name list.

!
- If the simulation input for your requests is not continuous, for example you are replaying a MAK Data Logger recording or rewinding a VR-Forces scenario, then you need to refresh the entity list and select the entity again before each request.
- The observer in RadarFX Server should not be attached to the selected entity.

   If you choose Set Location Manually, specify Latitude, Longitude, and Altitude.
4. For SAR, specify the down range in the Downrange box.

   Specify the cross range in the Crossrange box.
5. For ISAR, specify the view magnification in the View Magnification box.

Specifying the Power

The power value is the amount of power that the SAR system should use to generate the image. Higher power levels increase the image gain. Lower levels result in a darker image.

➤ To specify the power, enter a value in the Power box.

Requesting the Image

When you request an image, the client sends the request to the server. The server returns the image and it is displayed in the client window. ISAR returns multiple images.

➤ To request the image, click Send Image Request.

3.2.2. Requesting a Strip SAR Image

Strip SAR takes a series of images along a linear path. The client requests a SAR image at a time interval based on the speed of the entity and the width of the strip. The images are then combined in the client window. You can request a specific number of strips or continuous capture.

Figure 3-5 shows how strips might be configured along a path.

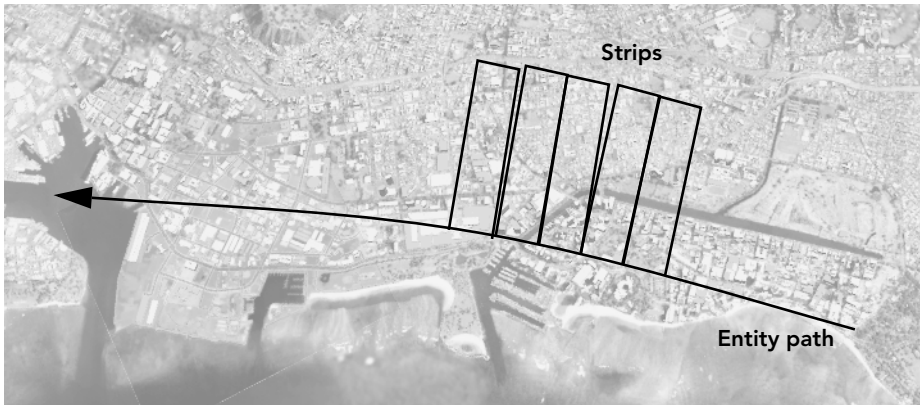

Figure 3-5. Strip SAR strips along a path

Figure 3-6 shows a Strip SAR image with five strips.

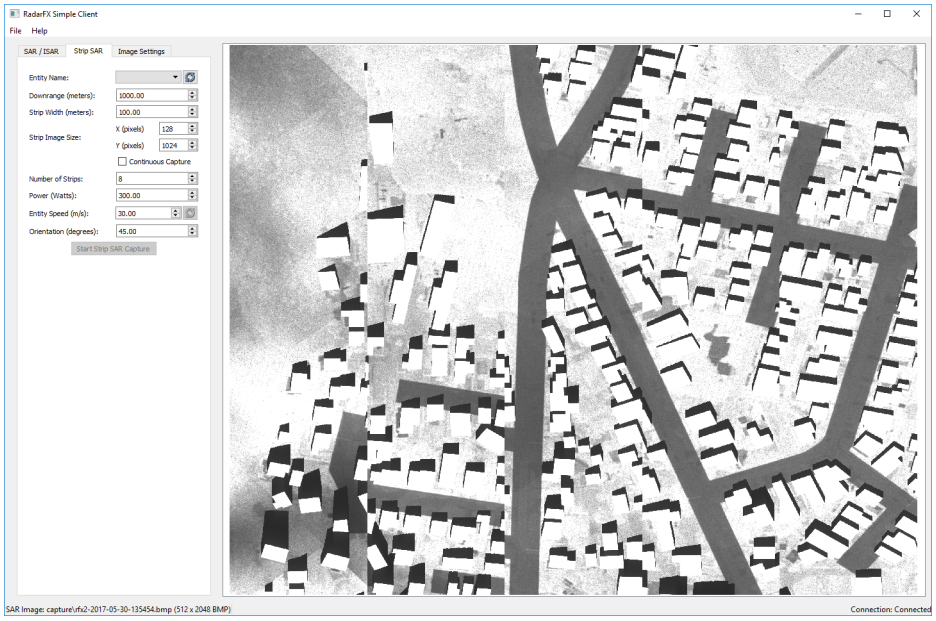

Figure 3-6. Strip SAR image with five strips

To request an image, you select an entity and specify the width of the strip and other parameters.

- When you request a Strip SAR image, the server takes an image to initialize the SAR sensor. Therefore, if you are saving the images, the server saves one more image than you requested. You can ignore the image with the earliest time stamp.
- While the server is taking images, it resizes the viewport to the correct size for the strip. When the SAR process is complete the viewport is not automatically resized to the full area of the window. To resize the viewport, resize the server window.
- For more information about setting up a client to handle strip SAR, please see *RadarFX Developers Guide*.

To request a Strip SAR image:

1. Select the Strip SAR tab.

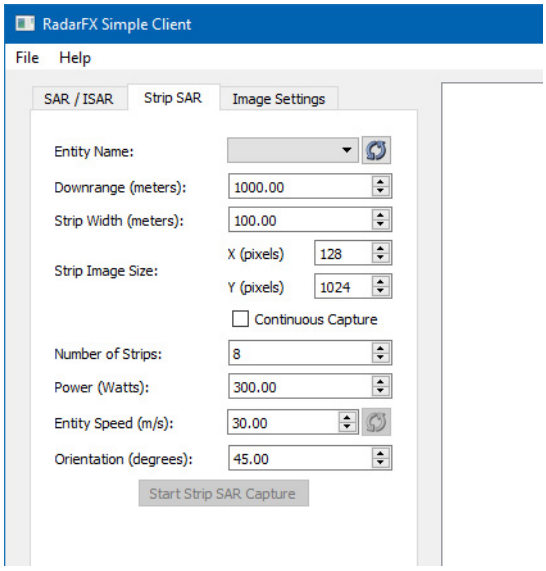

Figure 3-7. Strip SAR tab

2. Click Refresh button next to the Entity Name list.
3. In the Entity Name list, select the entity you want to follow.

- If the simulation input for your requests is not continuous, for example you are replaying a MAK Data Logger recording or rewinding a VR-Forces scenario, then you need to refresh the entity list and select the entity again before each request.
- The observer in RadarFX Server should not be attached to the selected entity.

4. Specify the Downrange, Strip Width, Number of Strips, and Power values.
5. Specify the Strip Image Size.
6. Optionally, for continuous capture, select the Continuous Capture check box.
7. Specify the speed of the entity. Click the Refresh button next to the speed value to get the entity's speed from the server.
8. Specify the orientation of the sensor. 0 degrees is out the left side of the entity; 90 degrees is straight down; 180 degrees is out the right side (Figure 3-8).

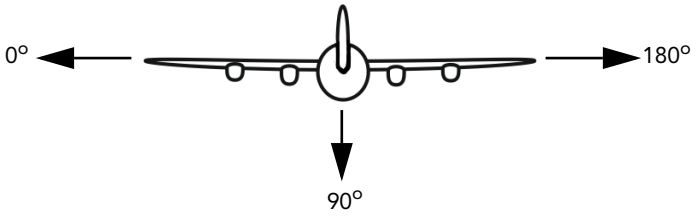

Figure 3-8. Strip SAR Orientation (rear view of plane)

9. Click Start Strip SAR Capture. If you select Continuous Capture, when the client receives the first image, the Start Strip SAR Capture button changes to read Stop Strip SAR Capture. Click the button when you want to stop capturing images.

3.3. Configuring Image Properties

You can specify the properties of the images that the server returns. You can also specify that the images be saved on the server. (The client does not save images automatically. For details about saving the client image, please see "Saving and Loading Images on the Simple Client," on page 3-12.) You can specify the following properties:

- Image size, in pixels. For strip SAR, the width is the final width of the image, not the width of each strip.
- Image Rotation. The image can be rotated so that North is up or the sensor's view is up.
- Image Format. Options are PNG, JPG, BMP, Raw, and NITF (National Imagery Transmission Format).
- Image Data Type. Byte, short, or float.

If you are requesting Strip SAR, the server saves an image for each strip.

To specify image properties:

1. In the Simple Client window, select the Image Settings tab (Figure 3-9).

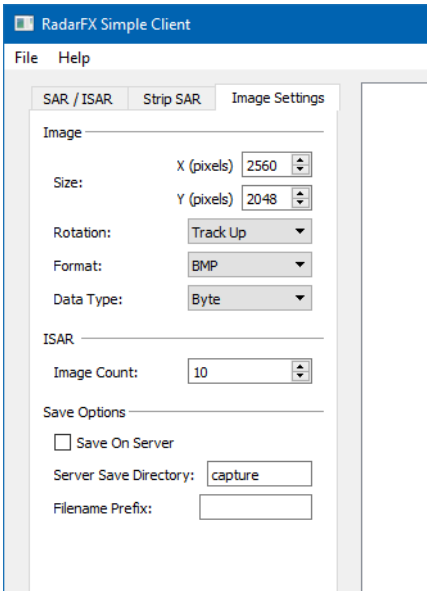

Figure 3-9. Image tab

2. In the Image section, change properties as desired.
3. For ISAR, specify the image count.
4. To save the images on the server, in the Save Options section, select Save On Server. Specify the directory in which to save images. The directory is relative to the directory where the executable is located. The default location is *./bin64/capture*.
5. Optionally, specify a prefix for the images.

3.4. Saving and Loading Images on the Simple Client

You can save the images you get from the server. You can load images that you have saved.

➤ To save an image, choose **File** → **Save Image**, and specify a name and location for the image.

➤ To load an image, choose **File** → **Load Image**, and select the image to load.

3.5. Creating a Configuration Preset

The client includes several preset configurations for the terrains that are shipped with RadarFX Server. The presets are stored in *./userData/radarFX/defaultClientPresets.json*. You cannot save configurations through the graphical user interface (GUI). The following is one of the presets included. To create your own, copy a block of code and change the values as desired.

```
{
  "name": "VR-Village-SAR",
  "ownshipLocation": {"lat": -0.014374, "lon": 0.001919, "alt": 1753},
  "targetLocation": {"lat": 0, "lon": 0, "alt": 0},
  "downrange": 400,
  "crossrange": 400,
  "power": 30,
  "sensor": "SAR (SensorFX)",
  "imageRotation": "North Up",
  "imageX": 1280,
  "imageY": 1024,
  "imageFormat": "PNG"
},
```

The Simple Client does not support presets for strip SAR.

3.5.1. Loading a Configuration Preset

➤ To load a preset, select a preset from the Preset Parameters list.

4. Adding GenesisMC Models to RadarFX

This chapter explains how to integrate materially classified models created with GenesisMC into RadarFX.

4.1. Introduction to Adding Models Processed by GenesisMC

GenesisMC is an application created by JRM Technologies that processes models and creates the files needed by RadarFX to render them based on material classification. This chapter assumes you have generated material classified versions of your models using GenesisMC and explains how to set up the files for RadarFX to use them.

4.2. Model File Setup

RadarFX uses the following types of files to load the materially classified version of a model:

- Material classification file (*.emat*).
- Material system file (*.ms*).
- Intensity modulation file (*.imod*) (optional).

These files are generated by GenesisMC.

RadarFX uses the names of the textures and models to look for the material classified configuration files. For example, the base set of files for the SH-60 SEA-HAWK (in *./data/Vehicles/RotaryWing/SH-60 SEA-HAWK*) before material classification is:

- DB_ROTOR_4PALES_128_rgba.meif
- DB_SH60L_1024_DESERT_CAMO_NML_rgb.meif
- DB_SH60L_1024_DESERT_CAMO_RFL_rgb.meif
- DB_SH60L_1024_DESERT_CAMO_rgb.meif
- DB_SH60L_1024_DESERT_CAMO_SPC_rgb.meif
- DB_SH60L_1024_rgb.meif
- DB_SH-60L_DESERT_CAMO_V7.0.medf
- DB_SH60L_FLOOR_256_rgb.meif
- DB_SH60L_INTERIOR_512_rgb.meif
- DB_SH60L_SHADOW_256_rgba.meif
- DB_SH60L_WINDOW_512_RFL_rgba.meif
- DB_SH60L_WINDOW_512_rgba.meif
- DB_SH60L_WINDOW_512_SPC_rgba.meif

4.2.1. The *.emat* Files

EMAT files are materially classified versions of the visual RGB textures. One of the RGB textures for the SH-60 is DB_SH60L_1024_rgb.meif. After processing the textures, GenesisMC creates three EMAT textures for *DB_SH60L_1024_rgb.meif – DB_SH60L_1024.emat1.dds*, *DB_SH60L_1024.emat2.dds*, and *DB_SH60L_1024.emat3.dds*. RadarFX does not use the emat1 file. In order to save texture memory, RadarFX uses the *.imod* file to fulfill the same purpose.

Copy the **.emat2.dds* and **.emat3.dds* files to the same directory as the RGB textures so that RadarFX can find them. The complete list of *.emat* files for the SH-60 is as follows:

- DB_ROTOR_4PALES_128.emat2.dds
- DB_ROTOR_4PALES_128.emat3.dds
- DB_SH60L_1024.emat2.dds
- DB_SH60L_1024.emat3.dds
- DB_SH60L_1024_DESERT_CAMO.emat2.dds
- DB_SH60L_1024_DESERT_CAMO.emat3.dds
- DB_SH60L_FLOOR_256.emat2.dds
- DB_SH60L_FLOOR_256.emat3.dds
- DB_SH60L_INTERIOR_512.emat2.dds
- DB_SH60L_INTERIOR_512.emat3.dds
- DB_SH60L_WINDOW_512.emat2.dds
- DB_SH60L_WINDOW_512.emat3.dds.

4.2.2. Material System File (*.ms*)

RadarFX uses the material system file to map a channel of the emat texture to a particular material. There might be multiple material system files in the GenesisMC output directory. The one that RadarFX needs ends in *Emat.ms*. The contents of the file will be nine lines that define six materials. In the SH-60 example, this file is *DB_SH-60L_DESERT_CAMO_V7.0Emat.ms*.

There are two ways to point to a material system file. The simplest way is to give the file the same name as the model file name minus the extension and add *Emat.ms*. This file must be in the same directory as the model file. In the SH-60 example, the model filename is *DB_SH-60L_DESERT_CAMO_V7.0.medf*. Therefore, the material system file is *DB_SH-60L_DESERT_-CAMO_V7.0Emat.ms*.

The other way to point to a material system file is in a model definition. For details, please see "The msFilename Parameter," on page 4-4.

4.2.3. Intensity Modulation File (*.imod*)

An intensity modulation file or *.imod* file is an optional file that defines how much of the RGB texture to mix with the Sensor response in the final result. It allows the intensity modulation to be defined on a per-material and per-sensor band level. The file must have the same name as the material system file, except that the extension must be *.imod* instead of *.ms*. This file must be in the same directory as the material system file. In the SH-60 example the material system file is *DB_SH-60L_DESERT_CAMO_V7.0Emat.ms* which corresponds to intensity modulation file *DB_SH-60L_DESERT_CAMO_V7.0Emat.imod*.

4.3. Model Definitions for Materially Classified Models

If you have generated material classification files for a model that is already configured in VR-Vantage, all you have to do is put the required files in the model directory as described in previous sections. You do not have to edit the model definition.

If you have created a new model, you must create a model definition, as described in Section 32.1, "Creating and Editing Model Definitions," in *VR-Vantage Users Guide*.

RadarFX supports an optional parameter, msFilename, that you can add to schemas and model definitions.

4.3.1. The msFilename Parameter

The msFilename parameter lets you point to a material system file that does not use the naming convention described in "Material System File (.ms)," on page 4-3. Possible reasons for this are if there are two models that need to use the same material system file or there are multiple variations of the same model that need different material systems.

The msFilename parameter is not available by default. You must be added to the model's schema before you can use it. Please see Section 31.5, "Adding a Parameter to a Schema," in *VR-Vantage Users Guide* for information about how to add a parameter to a schema.

The parameter must be called msFilename and have its type set to string.

To determine which schema to edit, look at the schema listed for the model definition for the model you are editing. The most likely schemas are Entity and Watercraft.

Index

I

L

M

O

P

Q

R

S

T

U

W

X-Y-Z

Link - Simulate - Visualize

150 CAMBRIDGE PARK DRIVE, 3RD FLOOR CAMBRIDGE, MA 02140 617.876.8085 www.mak.com