



MAK ONE



Copyright © 2024 MAK Technologies
All rights Reserved. Printed in the United States.

Under copyright laws, no part of this document may be copied or reproduced in any form without prior written consent of MAK Technologies.

VR-Exchange™, VR-TheWorld™, VR-Vantage™, DI-Guy™, and DI-Guy Scenario™ are trademarks of MAK Technologies. MAK Technologies®, VR-Forces®, RTIsPy®, B-HAVE®, and VR-Link® are registered trademarks of MAK Technologies.

GL Studio® is a registered trademark of The DiSTI® Corporation.

Portions of this software utilize SpeedTree® technology (©2008 Interactive Data Visualization, Inc.). SpeedTree is a registered trademark of Interactive Data Visualization, Inc. All rights reserved.

SilverLining™ is a trademark of Sundog Software.

Terrain Profiles are based in part on the work of the Qwt project (<http://qwt.sourceforge.net>).

3ds Max® is a registered trademark of Autodesk, Inc.

All other trademarks are owned by their respective companies.

For third-party license information, see "Third-Party Licenses" on page 9 in the Preface.

MAK Technologies
10 Fawcett Street, Suite 204
Cambridge, MA 02138 USA

Voice: +1 617.876.8085

Fax: +1 617.876.9208

info@mak.com

www.mak.com

Revision RFX-1.6-1-240426



Contents

Preface

MAK Products	6
How to Contact Us	8
Telephone	8
E-mail	8
Internet	8
Post	8
Document Conventions	8
Mouse Button Naming Conventions	9
Third-Party Licenses	9
Third-Party Libraries Supported	9
Library Licensing	10
Boost License	12
libXML and libICONV	13

1. Overview of RadarFX

1.1. Introduction to RadarFX	16
1.2. Installing RadarFX	17
1.2.1 Installation and Data Packages	17
1.2.2 Installing RadarFX on Windows	18
1.2.3 The makData Package	18
1.3. The RadarFX Directory Structure	20
1.4. Installing and Setting Up the MAK License Manager	21
1.4.1 Specifying the License Server	22
1.5. Installing an RTI	24
1.5.1 Installing the MAK RTI	24
1.6. Localizing the Graphical User Interface	25
1.6.1 Translating Other Interface Files	27

1.6.2 Applying the Language Files	28
2. Using RadarFX Server	
2.1. Starting RadarFX Server	30
2.2. Opening a Terrain	31
2.3. Connecting to a Simulation	31
2.4. Configuring RadarFX Server	31
2.4.1 Setting the Server Listen Port	32
2.5. Configuring Sensors	33
2.6. Using RadarFX with Geocentric Terrains	36
3. Using the RadarFX Simple Client	
3.1. Starting the Simple Client	38
3.2. Requesting an Image	38
3.2.1 Requesting a SAR or ISAR Image	39
3.2.2 Requesting a Strip SAR Image	42
3.3. Configuring Image Properties	45
3.4. Saving Images	46
3.5. Loading Images on the Client	47
3.6. Creating a Configuration Preset	48
3.6.1 Loading a Configuration Preset	48
4. Adding GenesisMC Models to RadarFX	
4.1. Overview of Adding Models Processed by GenesisMC	50
4.2. Model File Setup	50
4.2.1 The .emat Files	50
4.2.2 Material System File (.ms)	50
4.2.3 Intensity Modulation File (.imod)	51
4.3. Model Definitions for Materially Classified Models	51
4.3.1 The msFilename Parameter	51
Index	53



Preface

This manual is for people who will install RadarFX and use it to view simulations. The manual assumes that you are familiar with performing basic administrative tasks in your operating system.

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

MAK Products

The MAK Technologies 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 NETN FOM and 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 VR-Vantage and the VR-Vantage Toolkit.
 - VR-Vantage 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.
 - **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.
- **MAK Legion.** MAK Legion is a collection of applications and an SDK designed to support simulations with millions of simulation objects.

How to Contact Us

For 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
	Fax:	617-876-9208

E-mail

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

Internet

MAK website home page:	www.mak.com
License key requests:	www.mak.com/support/get-licenses
Technical support:	www.mak.com/mak-one/support/portal
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:	MAK Technologies 10 Fawcett Street, Suite 204 Cambridge, MA 02138 USA
--------------------------------	---

When requesting support, tell us the product you are using, the version, and the platform on which you are running. If you are using VR-Vantage, VR-Forces, VR-Engage, or DI-Guy, please also include your data compatibility version and date.

Document Conventions

This manual uses the following typographic conventions:

Sans Serif Bold	Indicates a key on the keyboard or menu options. Also indicates parameters.
Monospaced	Indicates values you enter such as commands or arguments.
<i>Monospaced</i>	Indicates command variables that you replace with appropriate values.
<i>Italic</i>	
{ }	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.
<i>Italic</i>	Indicates a file name, pathname, or a class name.
►	Indicates a 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 .
	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 *./RadarFX1.6/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.

Third-Party Libraries Supported

RadarFX 1.6 uses the indicated versions of the following libraries:

- VR-Link 5.9
- Qt 5.15.2 (VC 15 and Red Hat 8)
- OpenSceneGraph 3.6.4
- Boost 1.63.0
- FreeGlut 2.4.0 (Windows), 2.6.0 (Linux)
- GL Studio 8.2
- DI-Guy 13.7 (development version)
- Triton 4.43
- SilverLining 6.20
- SpeedTree 6.3.2
- CDB API 2.2 (Windows), 2.1 (Linux)

- osgEarth 3.2
- CUDA 11.7

Library Licensing

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

- CIGI
- Devil
- GEOS
- GStreamer
- LibIconV
- LibWebSockets
- OPCode
- OpenAL
- OSG
- OsgEarth
- Pthread
- Qt
- Qwt

The following libraries are covered by the ZLib license:

- Bullet Physics
- LibPNG
- LibSDL
- Minizip
- Zlib

The following libraries are covered by the MIT license:

- Expat
- FreeGLUT
- GDAL
- GeoTIFF
- GLM
- HarfBuzz
- LibCURL
- LibSquish
- LibXML
- LUA
- LuaBind
- Proj
- SMHasher
- TCLAP
- Qt Visual Graph Editor
- ZeroBrane Studio

The following libraries are covered by the BSD license:

- FreeType
- GLEW
- JPEG
- LibTiff
- Protobuf
- PCRE
- Ogg Vorbis
- WebP

The following libraries are covered by the commercial licenses:

- FLEXlm
- Gameware
- GLStudio
- JRM Technologies SenSimRT and SigSimRT
- OpenFlight
- SpeedTree
- Sundog Triton
- Sundog SilverLining
- 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:

SQLite is in the public domain.

SWIG is covered by the GPL license.

Boost License

VR-Link, and all MAK ONE products that use VR-Link, uses some code that is distributed under the Boost License. All header files that contain Boost code are properly attributed. The Boost website 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 ONE products that use 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, see the following websites:

- 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. Overview of RadarFX

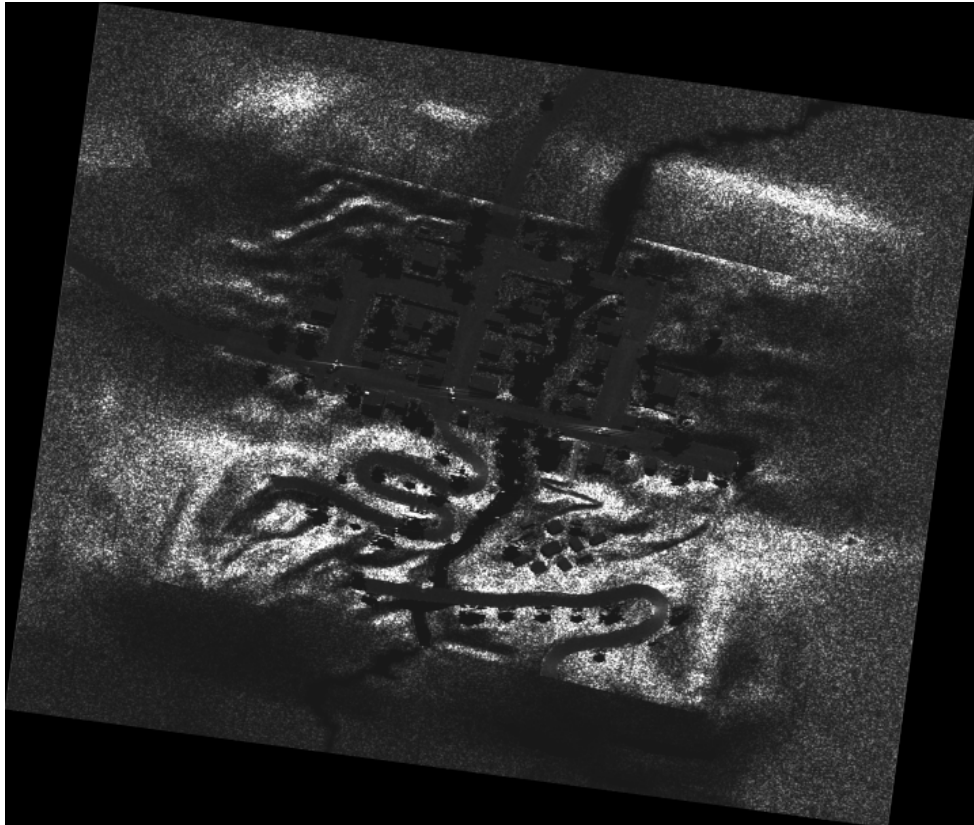
This chapter provides a brief introduction to RadarFX and explains how to install it. You must also install the MAK license manager. To use RadarFX with HLA, you must install an RTI.

1.1. Introduction to RadarFX	16
1.2. Installing RadarFX	17
1.2.1 Installation and Data Packages	17
1.2.2 Installing RadarFX on Windows	18
1.2.3 The makData Package	18
1.3. The RadarFX Directory Structure	20
1.4. Installing and Setting Up the MAK License Manager	21
1.4.1 Specifying the License Server	22
1.5. Installing an RTI	24
1.5.1 Installing the MAK RTI	24
1.6. Localizing the Graphical User Interface	25
1.6.1 Translating Other Interface Files	27
1.6.2 Applying the Language Files	28

1.1. Introduction to 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.

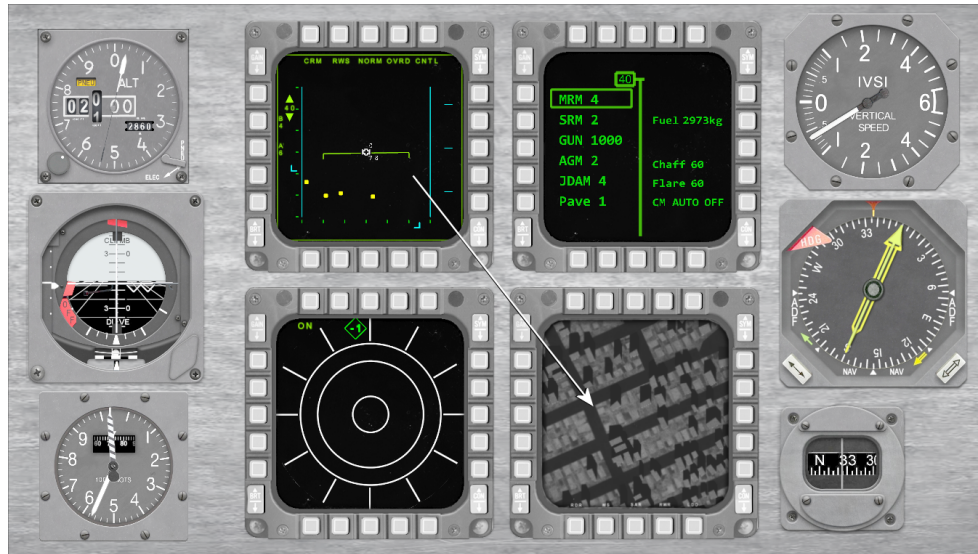
Figure 1. RadarFX SAR image



i RadarFX includes the *VR-Vantage Users Guide*, which explains how to use VR-Vantage and SensorFX in detail. 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 2).

Figure 2. SAR image



1.2. Installing RadarFX

1.2.1 Installation and Data Packages

MAK ONE installations are provided in one or more files (packages) that you can download from the MAK website.

There are four types of installation products for MAK ONE:

- Product installers – Each product has a single main installer for each operating system.
- Optional installers – Some products have installers for developer options.
- Required data packages – MAK ONE applications (VR-Vantage, VR-Forces, VR-Engage, SensorFX, and RadarFX) and the DI-Guy SDK have a collection of data packages that are required for those products' basic functions.
- Recommended data packages – Additional data packages (if available) are recommended to access the full capabilities of MAK ONE applications.

As of spring 2022, new releases of MAK ONE applications share the same data installation, therefore the same set of data packages is all you need for any or all applications (VR-Vantage, VR-Forces, VR-Engage, SensorFX, and RadarFX).

Dependencies

Licensing tools and dependent products have their own installers. For details, see "1.4. Installing and Setting Up the MAK License Manager" on page 21.

For interoperability using HLA, see "1.5. Installing an RTI" on page 24.

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

1.2.2 Installing RadarFX on Windows

Windows versions of RadarFX are provided as downloaded files.

- To install RadarFX, run the installer. Follow the instructions in the installation wizard. (You install the data used by RadarFX separately, as described in "1.2.3 The makData Package" below.)

Note the following:

- The installers are named to indicate the compiler used to build that version of RadarFX.
- 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 displays. This is due to how Windows scans setup programs before it executes them. If you experience this problem, turning off User Access Control can reduce or eliminate this delay.

Default Windows Installation Directory

The default root directory for installing MAK ONE products is *C:/MAK*.

You can use an environment variable, `MAK_INSTALL_DIR`, to specify a different location where all MAK ONE products and data should be installed by default. For example, if you define it as `MAK_INSTALL_DIR = D:\MyProject`, then all MAK ONE products automatically install to that location. The data packages and products would then install in that directory by default: *D:/MyProject/SharedData*, *D:/MyProject/vrlink5.9*, and so on.

Uninstalling from Windows

MAK ONE product installations include an uninstaller application (*unins000.exe*), which you can find in the installation directory. Run that program to uninstall the product.

When you run other installers for the product, such as the PDBs, there are additional uninstallers. If you run the uninstallers from the highest number down (for example, *unins002.exe* and then *unins001.exe*), running *unins000.exe* uninstalls everything else and removes the install folder.

When you uninstall VR-Vantage, VR-Forces, or VR-Engage, files such as scenarios or SMSs that reside in the product directory are also deleted. Before uninstalling, copy any of your custom files from the product directory to another folder.

The uninstaller does not remove the data package. That is in a separate directory (for example, *./SharedData/18/latest*).

 If you run only the product uninstaller, you can then simply delete the rest of the product's install folder. Note that if you do not run all of the uninstallers, running only the product uninstaller may not remove all of the shortcuts from the Windows Start menu.

1.2.3 The makData Package

The MAK Data Packages are used by multiple MAK ONE applications. The applications need to be able to find the data, wherever you install it. RadarFX looks for data in a default location or uses a collection of environment variables to find the data in the

location where you installed it. You have several options for installing the data and telling RadarFX where to look for it.

MAK Data Installation Directory

The shared data files are stored in a directory file hierarchy. This structure allows for multiple versions of data to be installed for use with different versions of RadarFX.

The directory looks like *shared-data-root/data-compatibility-version/stem/data-folders*, where:

- *shared-data-root* is the location of the shared data directory. The default is *C:/MAK/SharedData* on Windows and *~/MAK/SharedData* on Linux.
- *data-compatibility-version* is a number, such as 16 or 18.
- *stem* is the directory holding the data files, for example, *latest* for the most recent installation of that compatibility version or the date of a previous installation, such as *2022-11-13*.
- *data-folders* hold the various files that make up the MAK data files (models, audio files, fonts, and so on).

Advanced Windows users may want to install MAK Data Packages in a very specific way, for example, installing the data on a shared drive and configuring any machines with MAK ONE applications to look there for the data. Refer to the *MAK ONE Installation Guide* for instructions.

Installing the makData Package

► To install the data:

1. Install the MAK ONE Software Manager.
2. Run the MakOneSoftwareManager executable in the *.\bin64* folder.

On Windows 10 and 11, you can also open the Start menu and then choose **MAK Technologies > MAK ONE Software Manager**. (To launch MAK ONE products more easily than the Windows 10 Start menu, please see the *MAK ONE Installation Guide* or this blog entry: [Tech Tip: Starting MAK ONE Applications on Windows 10](#).)

3. Click Select and Install Data Packages in the welcome page.
4. Specify the directory where you saved the data files. The default is your Downloads folder, but you can click the folder icon to navigate to another folder.
5. Select the data files that you want to install.



MAK Technologies recommends that you install all of the data files if you have the space and are not absolutely sure which files you require.

You must install the Core data for RadarFX to function. To have access to other terrains and to see anything other than notional models, install the other data packages provided by MAK. If you do not install any other data, it is assumed that you are providing your own models and terrains.

6. Optionally, choose the location where you want to install the data.

MAK recommends that you install the files in the default location. While MAK ONE products allow for diverse installations, the typical use is the standard installation. If you require a more complex installation, refer to the *MAK ONE Installation Guide*.

7. If you have a previous installation of data for the same compatibility version, you are prompted to choose whether to back up the previously installed data, delete it, or overwrite it. See "Options When Installing a Data Package" below.
8. Click Install. The MAK ONE Software Manager shows the progress of the data installation.
9. When the installation is complete, click Exit.

Options When Installing a Data Package

When installing a data package using the MAK ONE Software Manager, if you have no preexisting installations of that compatibility version, it simply installs the data files in the `./SharedData/##/latest` folder (where `##` is the compatibility version).

If there is already a previous release of the same compatibility version installed, however, you are prompted to choose whether to back up, delete, or overwrite the data from the previous release.

- Backup. If you install a newer release and choose to create a backup the previous release, the files in the *latest* directory are moved to a new directory named for the data version's date in the same compatibility folder and the new files are then installed in the latest folder. For example, the old files move to `./SharedData/16/2022-11-13` and the new files are installed in `./SharedData/16/latest`.

Creating a backup is the recommended option in most circumstances. It allows all MAK ONE applications to use the newest data, while also giving you the option to revert to a previous release using command-line options or setting a Windows environment variable. See the *MAK ONE Installation Guide* for details.

- Delete. The installer deletes the old data from the *latest* directory and then installs the new data.
- Overwrite. The installer overwrites the previous data files in the *latest* directory with the newer ones you chose to install. It also adds files that are new and ignores files that were previously installed if the package you are installing does not include that file.

1.3. The RadarFX Directory Structure

The installation process creates the RadarFX directory tree shown in Figure 3 (runtime and toolkit).

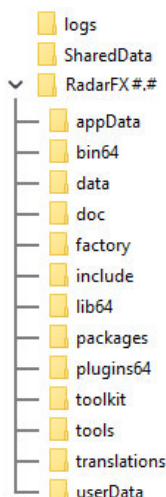
Figure 3. RadarFX directory structure

Table 1 describes the contents of the directories.

Table 1. Contents of RadarFX directories

Directory	Directory contents
<i>./appData</i>	Subdirectories contain data required by the RadarFX applications, such as model definitions and saved.
<i>./bin64</i>	Executables and dynamic link libraries.
<i>./data</i>	Data specific to the function of RadarFX.
<i>./doc</i>	RadarFX documentation and class reference files.
<i>./factory</i>	Default settings (for reverting to “factory”).
<i>./include</i>	Header files for the RadarFX Toolkit API.
<i>./lib64</i>	Libraries for building applications with the RadarFX Toolkit API.
<i>./packages</i>	Header files and libraries for third-party and supplementary APIs that developers may need to build against.
<i>./plugins64</i>	Plug-in shared libraries.
<i>./toolkit</i>	Examples for the RadarFX Toolkit API.
<i>./tools</i>	Helper applications such as the CIGI Host Emulator.
<i>./translations</i>	Translation files for localizing RadarFX menus.
<i>./userData</i>	Terrain, scene, and image files saved by application users.

1.4. Installing and Setting Up the MAK License Manager

Before you can use a MAK ONE 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 ONE products. You only need to install it on the computer that you will use as the license server.

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

The License Manager installer is included with MAK installation media. It is separate from the product installers. You can download the installers from the [MAK ONE Product Downloads](https://www.mak.com/mak-one-downloads) page at <https://www.mak.com/mak-one-downloads>.

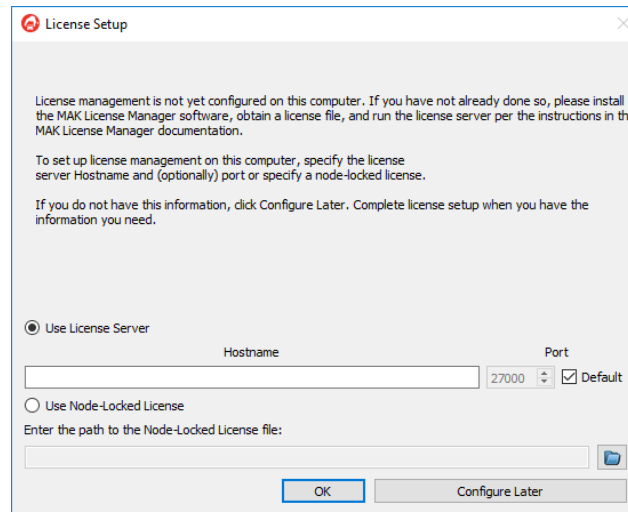
Complete installation and configuration instructions are included with the License Manager installer. The *MAK ONE Installation Guide* is available at <https://mak.com/mak-one-installation-guide>.

Some customers use dongle licenses instead of running a license server. Instructions for using dongles are also in the *MAK ONE Installation Guide*.

1.4.1 Specifying the License Server

The first time you run a MAK ONE product on a particular computer, the License Setup dialog box opens (Figure 4). It prompts you to enter the hostname of the license server and, optionally, a port number.

Figure 4. 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 ONE products until you set up license management.

If you know the hostname, enter 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 ONE 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 ONE product.
- If you have changed the license server, you may want to clear the license cache. 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*.) You can also clear the cache in the RTI Assistant.

Setting 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 or node-locked license 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.

For license servers, 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 ONE products run on.

Windows

► To set 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 displays.
3. Select *Edit the System Environment Variables*. The System Properties dialog box opens to the Advanced tab.
4. Click the Environment Variables button. The Environment Variables dialog box opens.
5. Below the System Variables list, 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 save out of each dialog box and set the variable.

Linux

On Linux, you set environment variables in your `.cshrc` (or equivalent startup file).

- To specify a license server, set the variable similarly to:

```
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:

```
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 ONE 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. MAK Technologies has the MAK RTI, but MAK ONE products do not require that you use the MAK RTI.

i 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 ONE products, please visit the Product Specifications & Dependencies page (<https://www.mak.com/mak-one-specs>).

The following instructions explain how to install the MAK RTI. If you are using an RTI provided by another vendor, please refer to that RTI's documentation for installation and setup instructions.

1.5.1 Installing the MAK RTI

To install the MAK RTI, follow the instructions in Chapter 2 of *MAK RTI Users Guide*. To install another RTI, see the vendor's documentation.

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 ONE product with the MAK RTI:

1. Ensure that the license server is running.
2. Start the application. The RTI Assistant prompts 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.

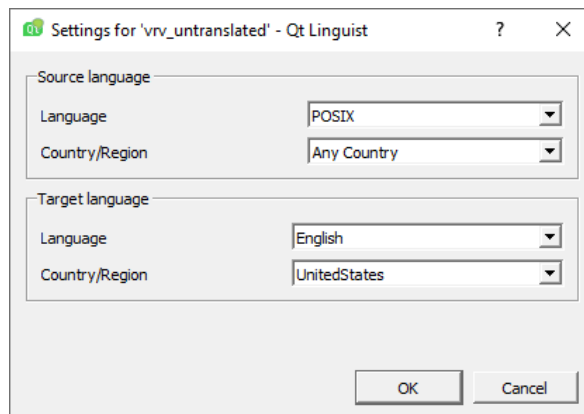
1.6. Localizing the Graphical User Interface

You can localize your MAK ONE 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:

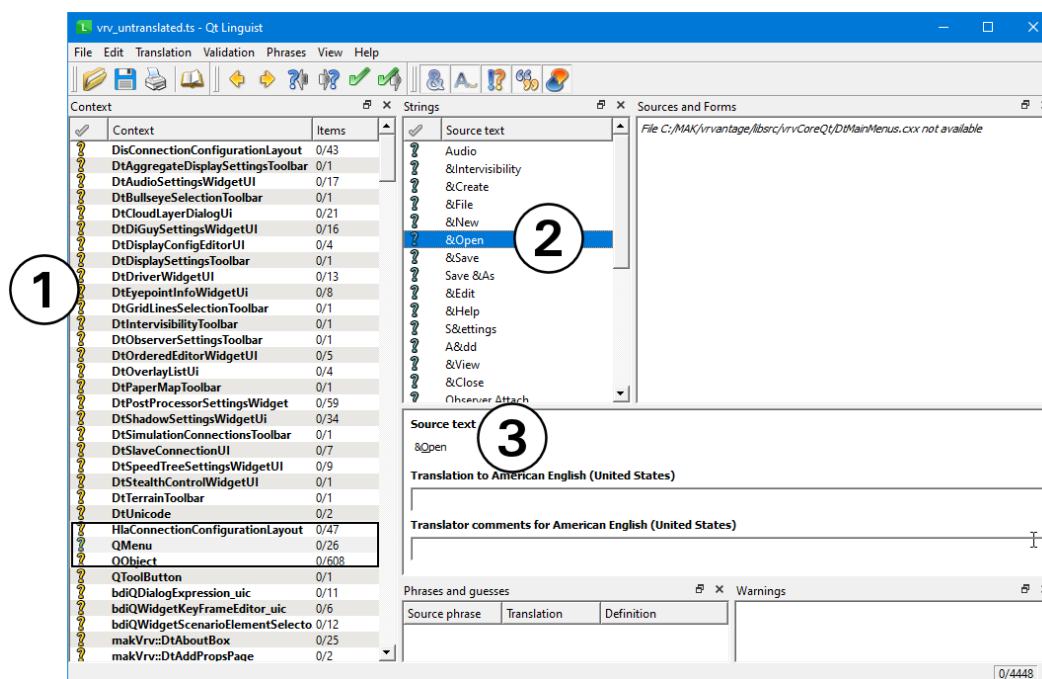
1. Start the Qt Linguist utility. It is *./translations/linguist.exe*
2. In the Qt Linguist window, choose **File > Open**. Select the file *./translations/product_untranslated.ts*. The Settings for filename dialog box opens (Figure 5).

Figure 5. Linguist settings dialog box



3. Select values for the source language, target language, and target region or country.
4. Click OK. The translation file opens and a list of classes displays on the left side of the window (callout 1 in Figure 6).

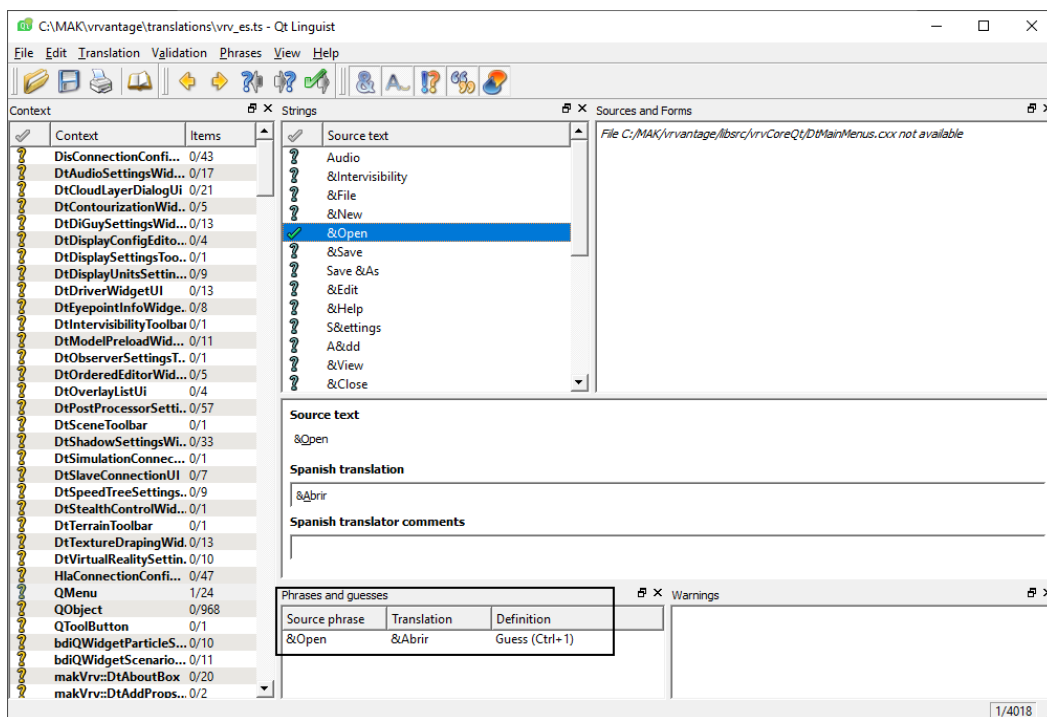
Figure 6. Linguist window



5. To translate text, 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 only need to translate the text. In Figure 6, **QMenu** is selected. The Strings window lists text that can be translated (callout 2). Text that has not been translated has a blue question mark in the checkmark column.
6. In the Source Text list, select the word you want to translate. The text displays under Source Text in the translation area (callout 3 in Figure 6).

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 7). 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 7. 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 the 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	30
2.2. Opening a Terrain	31
2.3. Connecting to a Simulation	31
2.4. Configuring RadarFX Server	31
2.4.1 Setting the Server Listen Port	32
2.5. Configuring Sensors	33
2.6. Using RadarFX with Geocentric Terrains	36

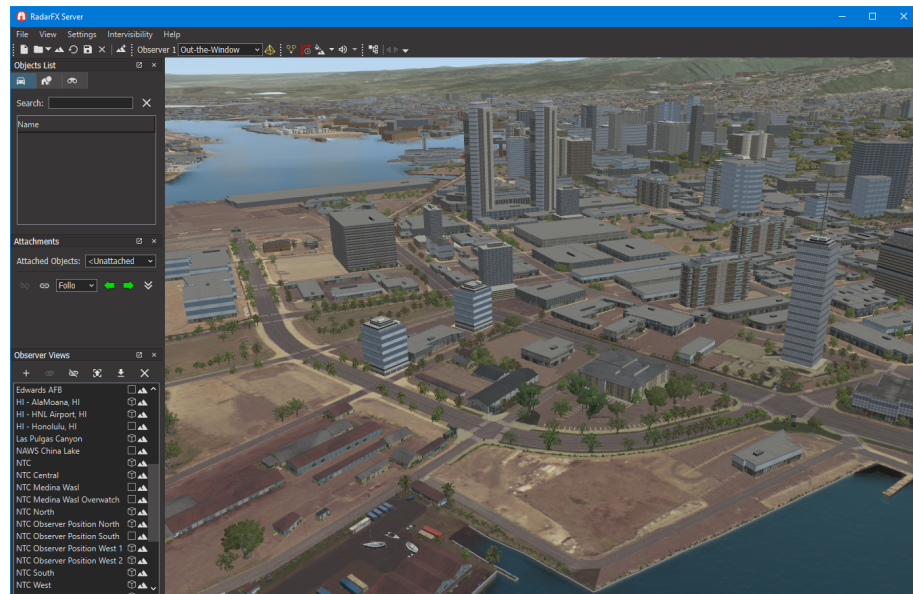
2.1. Starting RadarFX Server

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

► **To start RadarFX:**

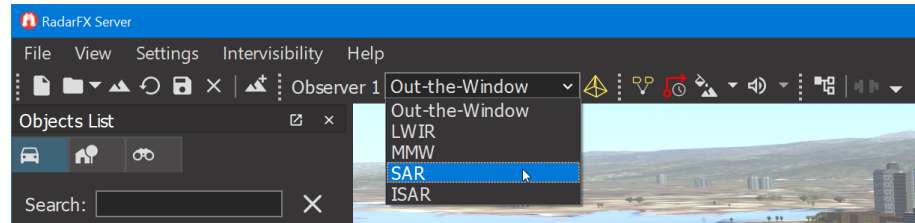
1. On the Start menu, choose **MAK Technologies > RadarFX Server**.
RadarFX Server starts and the Load Terrain dialog box opens.
2. Select a terrain to load. RadarFX Server loads the terrain and displays it in the window (Figure 8).

Figure 8. 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 the SensorFX section of the *VR-Vantage Users Guide* and the chapters on material classification in the *Adding Content to MAK ONE Applications Reference Manual*. There are several materially classified terrains, including *VR-Village.mtf*, that are part of the MAK ONE data package.
- RadarFX does not support geocentric terrains if they are part of a terrain that does not include an osgEarth terrain patch. For more information, see "2.6. Using RadarFX with Geocentric Terrains" on page 36.

3. If you plan to capture SAR, ISAR, or MMW images using the RadarFX Simple Client or a similar client, set the observer mode to match your image type. In the Observer list, choose MMW, SAR, or ISAR (Figure 9).

Figure 9. Setting the observer mode

i Setting the observer mode to match your image type reduces the time it takes to generate an image.

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, see the section "Opening a Terrain" in the *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 the section "Connecting to a Simulation Network" in the *VR-Vantage Users Guide*.

2.4. Configuring RadarFX Server

You can configure these settings for RadarFX Server:

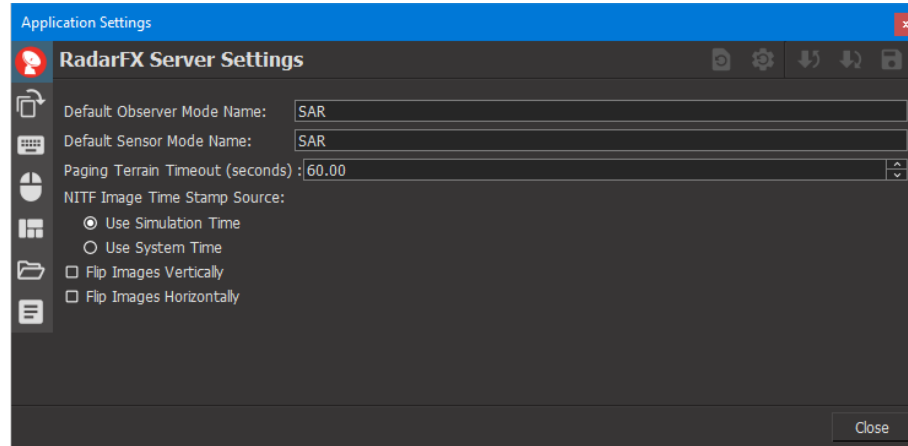
- **Server Listen Port.** The port that the server is listening on for incoming connections from clients. Changing this value restarts the server immediately.
- **Default Observer Mode Name.** The default observer mode to use for image captures.
- **Default Sensor Mode Name.** The default sensor mode 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, see "3.3. Configuring Image Properties" on page 45.)
- **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 10).

Figure 10. RadarFX Server Settings page



3. Change parameter values as desired.

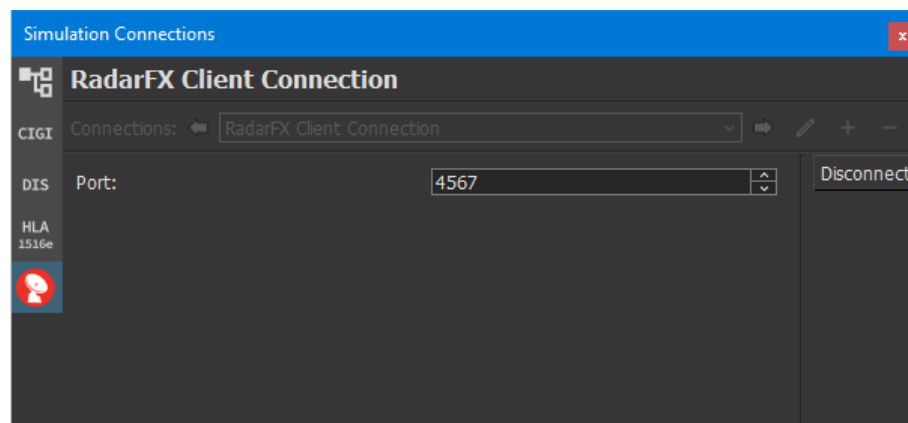
2.4.1 Setting the Server Listen Port

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 11).

Figure 11. 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 more details about sensor configuration, see "Configuring SensorFX Sensors" in the SensorFX section of the *VR-Vantage Users Guide*:

- The options on the Effects tab are the same for SAR and MMW as they are for LWIR.
- The options on the Advanced tab are the same for all sensor modes.
- When you open the Display Settings dialog box, the Sensor Mode value (at the top of the Sensor Settings page) changes to correspond to the observer mode setting for the main window.

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 on page 36.

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

Table 2. 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 atmospheric, 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 Downrange	(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 Crossrange	(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.

Table 2. Sensor parameters for SAR, ISAR, and MMW sensors (continued)

Parameter	Description
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.
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 $1/\text{PRF}$. 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 $1/\text{PW}$ and $c/2\text{DR}$, 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. Sensor parameters for SAR, ISAR, and MMW sensors (continued)

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 $c/(2\delta f)$, 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. Sensor parameters for SAR, ISAR, and MMW sensors (continued)

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 atmospheric, 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	38
3.2. Requesting an Image	38
3.2.1 Requesting a SAR or ISAR Image	39
3.2.2 Requesting a Strip SAR Image	42
3.3. Configuring Image Properties	45
3.4. Saving Images	46
3.5. Loading Images on the Client	47
3.6. Creating a Configuration Preset	48
3.6.1 Loading a Configuration Preset	48

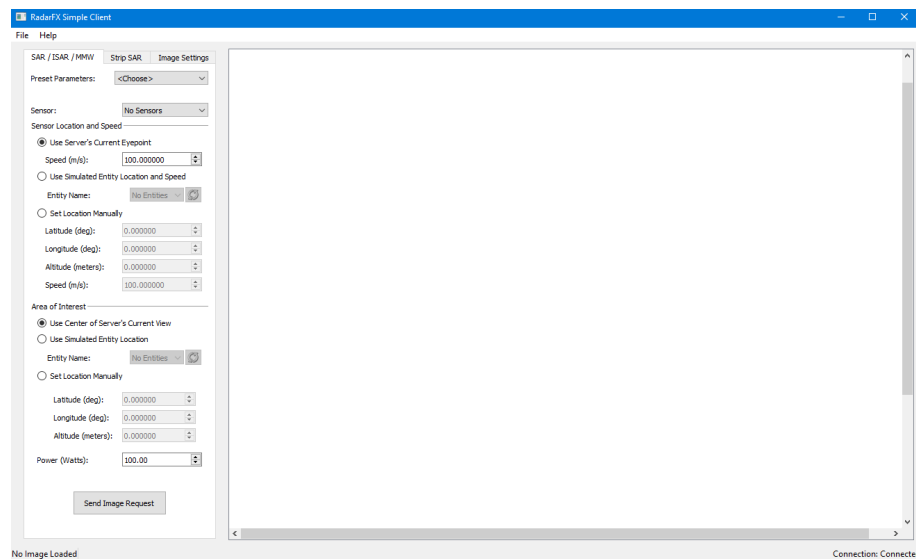
3.1. Starting the Simple Client

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 12).

Figure 12. 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 13. SAR image

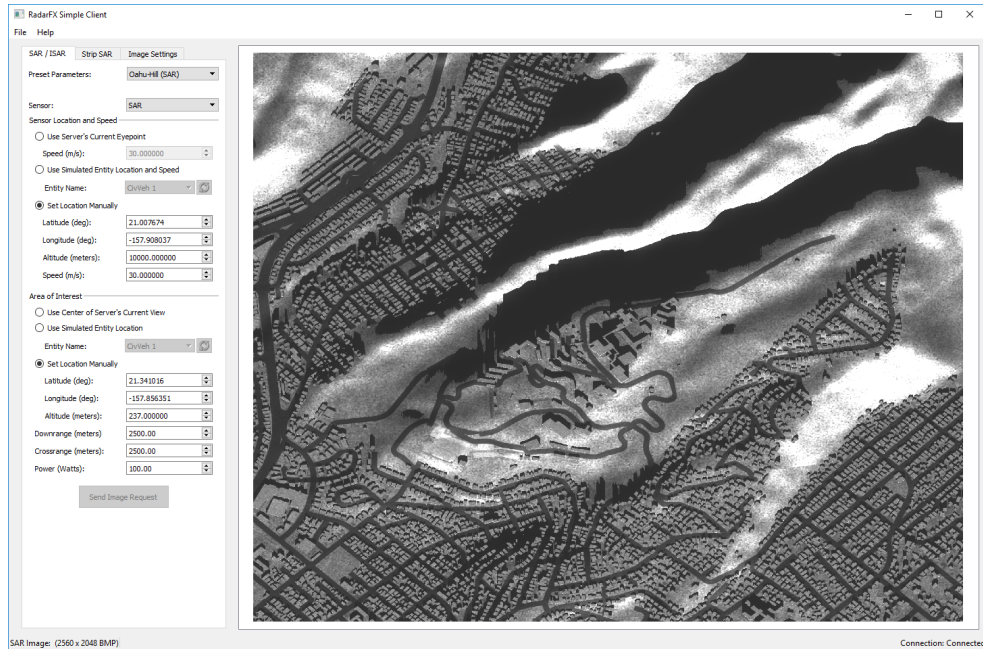
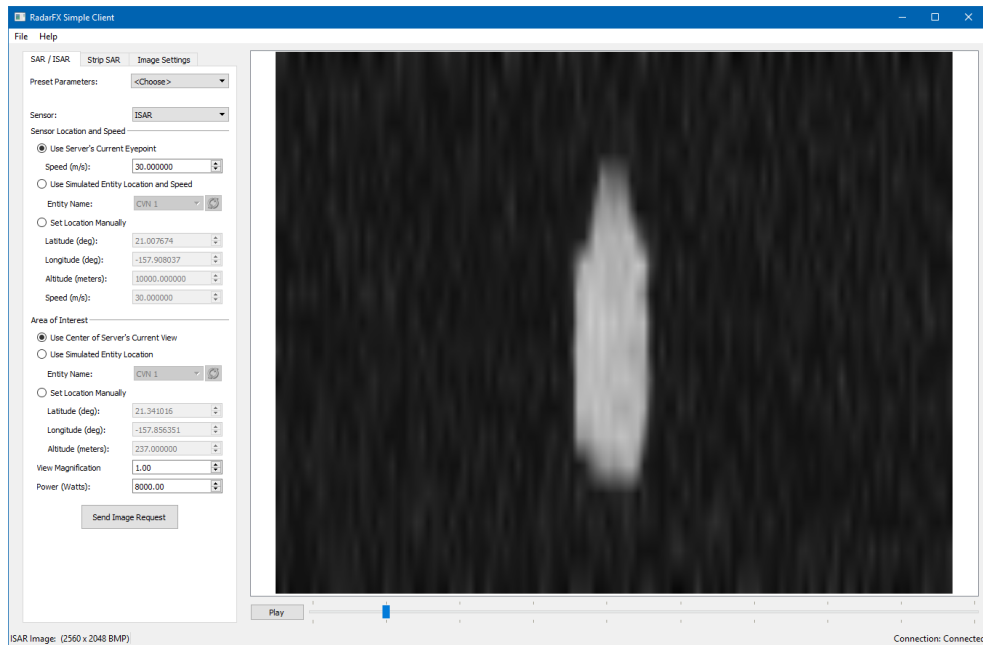


Figure 14. ISAR image



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, see "3.6. Creating a Configuration Preset" on page 48.

i 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/MMW tab.
2. Specify the sensor to use. For details, see "Specifying the Sensor" below.
3. Specify the location of the sensor. For details, see "Specifying the Location of the Sensor" below.
4. Specify the area of interest. For details, see "Specifying the Area of Interest" on the facing page.
5. For SAR, specify the downrange and crossrange values. For details, see "Specifying the Area of Interest" on the facing page.
For ISAR, specify the view magnification, integration time, and power.
6. Set the power to use. For details, see "Specifying the Power" on page 42.
7. Optionally, change Image settings. For details, see "3.3. Configuring Image Properties" on page 45.
8. Send the request. For details, see "Requesting the Image" on page 42.

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 the desired 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.

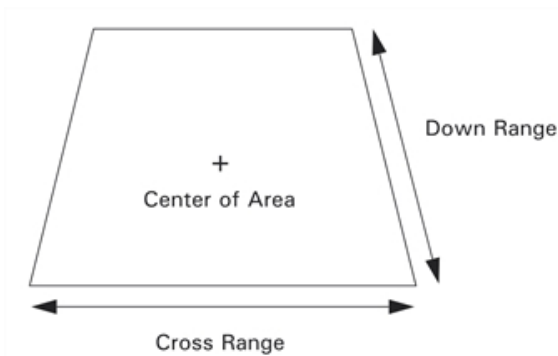
Specifying the Area of Interest

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 you specify by 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 15.

Figure 15. Area of interest



i 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.

3. For SAR, specify the down range in the Downrange box, and the cross range in the Crossrange box.

For ISAR, specify the view magnification and integration time.

Specifying the Power

The power value is the amount of power, in watts, that the SAR or ISAR 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 16 shows how strips might be configured along a path.

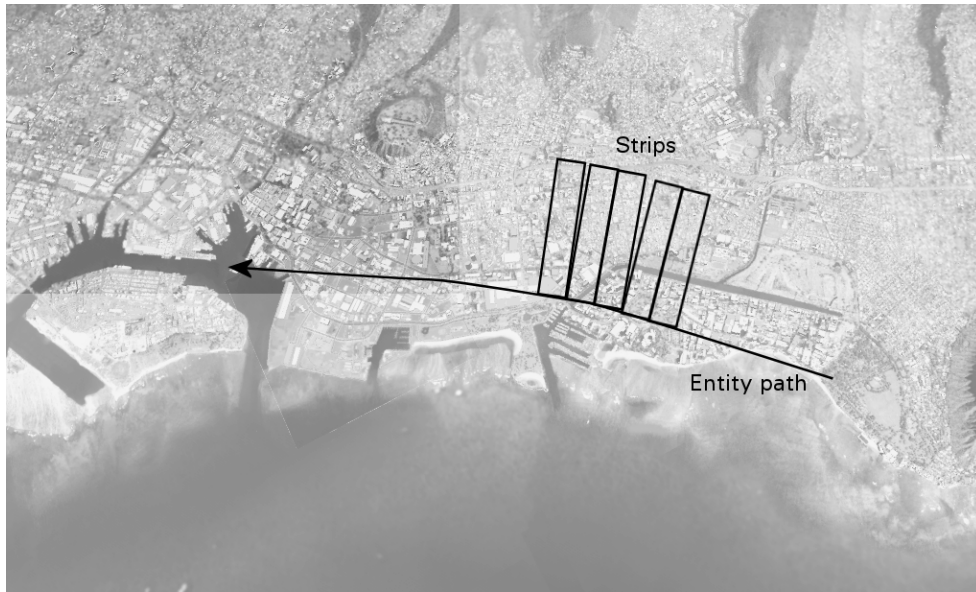
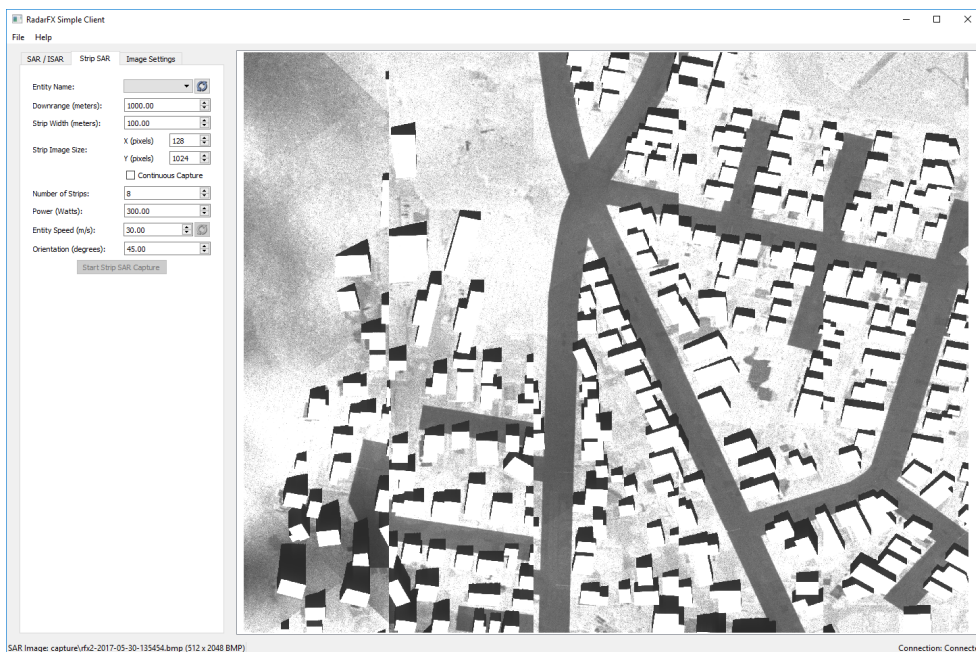
Figure 16. Strip SAR strips along a path

Figure 17 shows a Strip SAR image with five strips.

Figure 17. 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, see *RadarFX Developers Guide*.

► **To request a Strip SAR image:**

1. Select the Strip SAR tab.

Figure 18. 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.

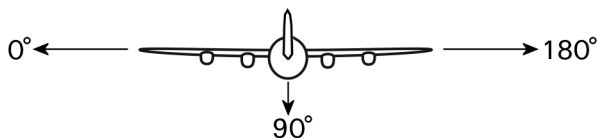


- If the simulation input for your requests is not continuous, for example you are replaying a MAK Data Logger recording or rewinding a VR-Forces scenario, then you need to refresh the entity list and select the entity again before each request.
- The observer in RadarFX Server should not be attached 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 19).

Figure 19. 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, see "3.4. Saving Images" on the next page.)

You can specify these image properties:

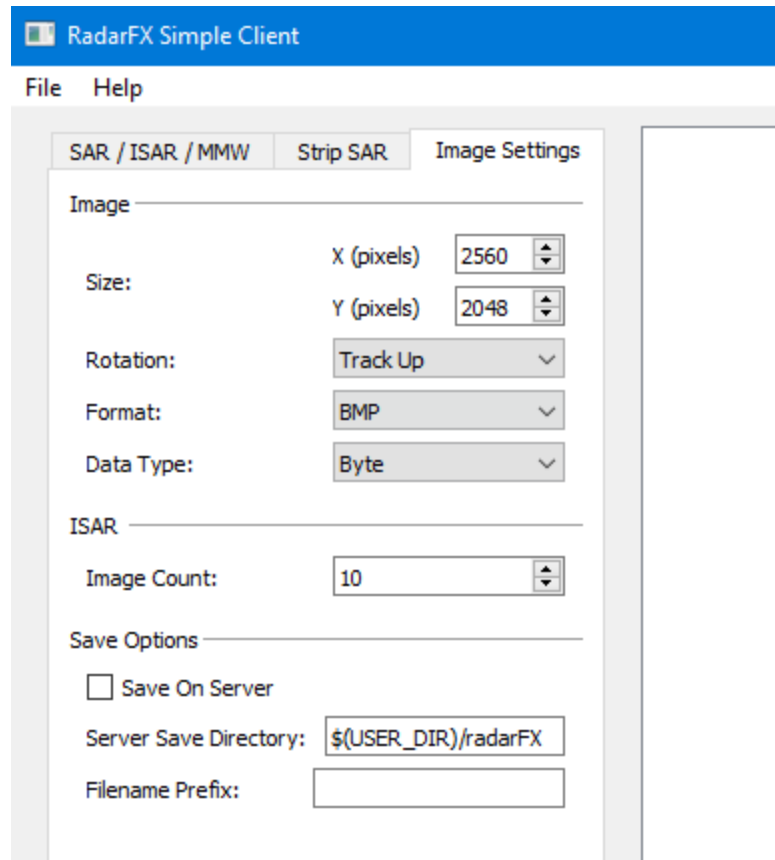
- Size, in pixels. For strip SAR, the width is the final width of the image, not the width of each strip.
- Rotation. The image can be rotated so that North is up or the sensor's view is up.
- Format. Options are PNG, JPG, BMP, Raw, NITF (National Imagery Transmission Format, version 2.1), Raw_RG and GTiff (Geographic Tagged Image File Format, or GeoTIFF).
- 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 20).

Figure 20. Image tab



2. In the Image section, change properties as desired.
3. For ISAR, specify the image count.
4. Optionally, specify a prefix for the image names.

3.4. Saving Images

You can save the images you get from the server in a variety of formats. (RadarFX uses NITF version 2.1, among other formats.)

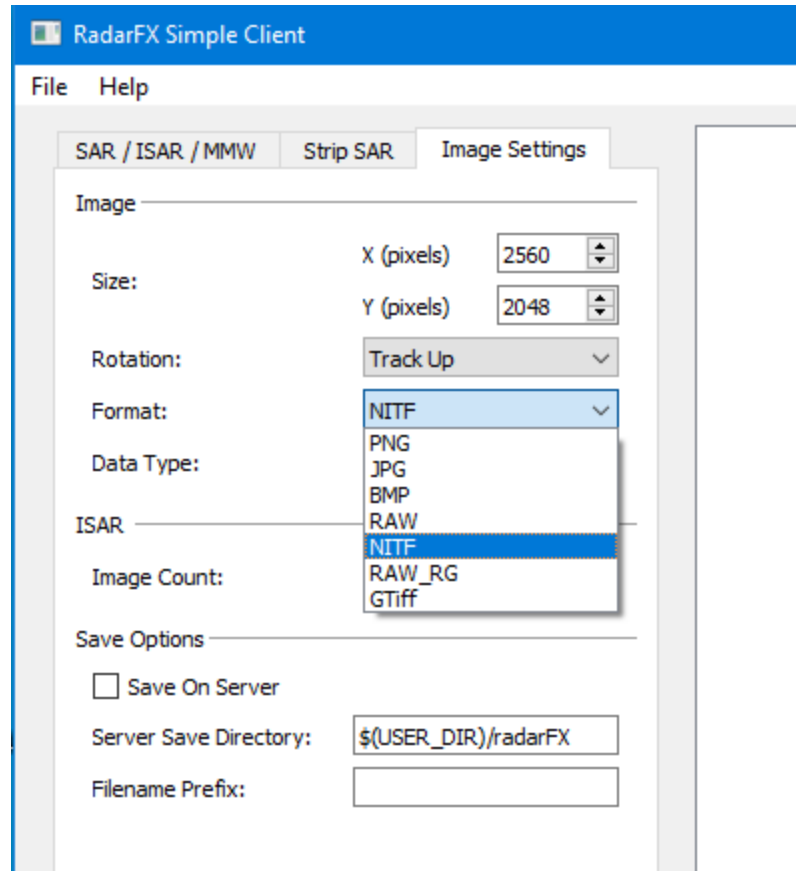
- To save an image in BMP, JPG, or PNG format, choose **File > Save Image** and specify a name and location for the image.

⚠ Do not use **File > Save Image** to capture GeoTIFF, NITF, raw, or raw_RG images.

► **To save a GeoTIFF, NITF, raw, or raw_RG image:**

1. Launch both RadarFX Server and your RadarFX client.
2. In the RadarFX Simple Client, switch to the Image Settings tab. (For details on image settings not covered here, see "3.3. Configuring Image Properties" on page 45.)

Figure 21. The Image Settings tab in the RadarFX Simple Client.



3. From the Format list, select GTiff, NITF, RAW, or RAW_RG.
4. Select Save on Server.
By default, RadarFX Server saves images in a subdirectory of its user directory, `./userData/radarFX`.
5. Optionally, to change where images are saved, enter a new directory name in the Server Save Directory field.
6. Send an image request from the client. (In the simple client, use the SAR / ISAR / MMW or Strip SAR tab.)
7. Look for the image file in the directory specified in the Server Save Directory field.

3.5. Loading Images on the Client

You can load BMP, JPG, or PNG images that you have saved. The client cannot load GeoTIFF, NITF, raw, or raw_RG images.

- To load an image, choose **File > Load Image** and select the image to load. By default, the simple client looks for images in `./userData/radarFX`.

3.6. 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 included presets. To create your own, copy a block of code and change the values as desired. Separate each preset code block with a comma.

```
{
  "name": "VR-Village (SAR)",
  "currentEyepointSpeed" : 30,
  "ownshipLocation": {"lat": -0.081294, "lon": 0.040931, "alt":
5000},
  "ownshipSpeed" : 30,
  "targetLocation": {"lat": -0.000102, "lon": 0.000175, "alt":
0.001},
  "downrange": 300,
  "crossrange": 300,
  "power": 100,
  "sensor": "SAR",
  "imageRotation": "Track Up",
  "imageX": 2560,
  "imageY": 2048,
  "imageFormat": "BMP"
},
```

 The Simple Client does not support presets for strip SAR.

3.6.1 Loading a Configuration Preset

- To load a preset, select a preset from the Preset Parameters list on the SAR / ISAR / MMW tab.



4. Adding GenesisMC Models to RadarFX

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

4.1. Overview of Adding Models Processed by GenesisMC	50
4.2. Model File Setup	50
4.2.1 The .emat Files	50
4.2.2 Material System File (.ms)	50
4.2.3 Intensity Modulation File (.imod)	51
4.3. Model Definitions for Materially Classified Models	51
4.3.1 The msFilename Parameter	51

4.1. Overview of 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.

For more information, refer to the chapter on model material classification in the *Adding Content to MAK ONE Applications Reference Manual*.

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.

4.2.1 The *.emat* Files

EMAT files are materially classified versions of the visual RGB textures. After processing the textures, GenesisMC creates three EMAT textures for the *.meif* file: *.emat1.dds*, *.emat2.dds*, and *.emat3.dds*. RadarFX does not use the *emat1* file. 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 to enable RadarFX to find them.

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, see "4.3.1 The *msFilename* Parameter" on the facing page.

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.

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 the section "Creating and Editing Model Definitions" in the *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 "4.2.2 Material System File (.ms)" on the previous page. 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. It must be added to the model's schema before you can use it. For instructions, see the section "Adding a Parameter to a Schema" in the *Adding Content to MAK ONE Applications Reference Manual*.

The parameter must be called **msFilename** and its type must be 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

A

- area of interest
 - specifying, 41
- Automatic Gain Control, 33

C

- center frequency, 34
- configuration
 - MAK RTI file, 24
 - preset, 48
- configuration preset in simple client
 - loading, 48
- configuring
 - image properties, 45
 - RadarFX Server, 31-32
 - sensors, 33
- connecting
 - simulation to, 31
- cross range
 - region, 33

D

- data
 - install, 17
 - options, 20
- directivity
 - receiver, 35
 - transmitter, 34

- directory
 - Windows default install, 18
- display type, 36
- Doppler coloring, 33
- Doppler Spatial Effects, 34
- downrange
 - region, 33

E

- emat file, 50
- emitter
 - power, 35
- environment variable
 - MAK_INSTALL_DIR, 18
 - MAKLMGRD_LICENSE_FILE, 23
 - RTI_CONFIG, 25
- executable
 - installation location
 - Windows, 18

F

- FDD file, 24
- FED file, 24
- file
 - FDD, 24
 - FED, 24
 - RID, 24
- FLEXIm, 21
- FOM, 24
- formats for images, 45
- frequency
 - center, 34

G

gain
 intensity modulation, 36
 pattern, 35
GenesisMC, 50
GUI
 localizing, 25

H

hostname
 license server, 22

I

image
 formats, 45
 loading on simple client, 47
 properties
 configuring, 45
 requesting, 42
 requesting SAR or ISAR, 39
 saving on simple client, 46
 Strip SAR
 requesting, 42
image orientation, 33
imod file, 50-51
installation, 17
 data
 options, 20
 default directory, Windows, 18
 RTI, 24
 Windows, 18
integration path length, 35
Integration Time, 35
intensity modulation file, 50-51
intensity modulation gain, 36
ISAR
 sensor, 33

L

license manager, 21
license server
 hostname, 22
loading
 configuration presets in simple client, 48
 images on simple client, 47
 terrain, 30

localization
 RadarFX GUI, 25

M

MAK data
 install
 options, 20
MAK RTI configuration, 24
MAK_INSTALL_DIR environment variable, 18
MAKLMGRD_LICENSE_FILE environment variable,
 23
material classification file, 50-51
material system file, 50
materially classified model
 files for, 50
MMW
 sensor, 33
model
 materially classified
 files for, 50
model definition, 51
ms file, 50
msFilename parameter, 51

O

observer mode
 default, 31
orientation
 image, 33

P

parameter
 msFilename, 51
path
 default Windows install, 18
 MAK RTI, 24
pattern
 gain, 35
polarization angle
 receiver, 35
 transmitter, 34
port
 server listen, 31-32
power
 emitter, 35
 specifying, 42
preset
 configuration, 48

- product installation, 17
 - Windows, 18
- properties
 - image, 45
- pulse repetition frequency, 34
- pulse width, 34

Q

- Qt Linguist, 25

R

- RadarFX
 - starting server, 30
- RadarFX Simple Client
 - starting, 38
- receiver
 - directivity, 35
- receiver polarization angle, 35
- region
 - cross range, 33
 - downrange, 33
- requesting
 - image, 42
 - SAR or ISAR image, 39
 - Strip SAR image, 42
- RID file, 24
- rid.mtl file, 24
- RTI, 24
- RTI_CONFIG environment variable, 25

S

- SAR
 - sensor, 33
- SAR/ISAR/MMW tab in simple client, 48
- saturation
 - Signal-to-Noise ratio, 35
- saving
 - images on simple client, 46
- sensor
 - configuring, 33
 - ISAR, 33
 - location
 - specifying, 40
 - MMW, 33
 - SAR, 33
 - specifying, 40
- sensor mode
 - default, 31

- server
 - configuring, 31-32
 - starting, 30
- server listen port, 31-32
- Signal-to-Noise ratio
 - saturation, 35
- simple client
 - using, 38
- simulation
 - connecting to, 31
- specifying
 - area of interest, 41
 - power, 42
 - sensor, 40
 - sensor location, 40
- starting
 - RadarFX Simple Client, 38
- Strip SAR
 - image
 - requesting, 42
- synthetic-aperture radar
 - SAR, 16
- system
 - temperature, 35

T

- temperature
 - system, 35
- terrain
 - loading, 30
 - materially classified, 30
- translation, RadarFX GUI text, 25
- transmitter directivity, 34
- transmitter polarization angle, 34

U

- using
 - simple client, 38

V

- variable, environment, 23

W

- Windows
 - default install directory, 18
 - product installation, 18



XML

RTI configuration file, 24



RFX-1.6-1-240426