



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 10 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 DIG-13.7.1-10-240819



Contents

Preface

Documentation	6
Try the Help	6
Derivative Work	6
MAK Products	6
How to Contact Us	8
Telephone	9
E-mail	9
Internet	9
Post	9
Document Conventions	9
DI-Guy Conventions	10
Mouse Button Naming Conventions	10
Third-Party Licenses	10
Third-Party Libraries Supported	10
Library Licensing	11
Boost License	13
libXML and libICONV	14
Lua	14
Scintilla and SciTE	14

1. Introduction to DI-Guy

1.1. Introduction	18
1.2. How Is DI-Guy Used?	18
1.2.1 Expressive Faces	19
1.2.2 Vehicle Characters	19
1.3. High-Performance Real-time Motion Engine	19
1.4. DI-Guy Works in any Rendering Environment	20

1.5. Networking with DI-Guy	20
-----------------------------------	----

2. Installing DI-Guy

2.1. Introduction	22
2.2. Installing DI-Guy on Windows	22
2.2.1 Installing DI-Guy in Silent Mode	22
2.2.2 The DI-Guy Environment Variable	22
2.2.3 Managing Multiple Versions of DI-Guy on Windows	23
2.3. Installing DI-Guy on Linux	23
2.4. Installing DI-Guy PDB Files for Debugging Purposes	24
2.4.1 Installing PDB Files on Windows	25
2.4.2 Reporting an Issue Using the .callstack.log and .dmp Files	25
2.5. DI-Guy Directory Structure	26
2.6. Installing and Setting Up the MAK License Manager	27
2.6.1 Specifying the License Server	27
2.7. Troubleshooting Performance and Display Issues	29

Index	31
--------------------	-----------



Preface

This manual is written for people who want to install and use DI-Guy products. The manual assumes that you are familiar with your operating system and that you know how to work in your computer's graphical window environment.

Please see release documentation for information about changes and updates to DI-Guy since this manual was completed.

Documentation

DI-Guy comes with the following documentation:

- *DI-Guy SDK Developers Guide*. Online documentation that lists all the functions and classes of DI-Guy SDK and explains how to program using the DI-Guy SDK.
- *DI-Guy Documentation Center*. Includes HTML versions of all of the books listed below, allowing you to search the entire documentation set. It also includes links to developer documentation and the PDF versions of the guides.
- *DI-Guy Getting Started Users Guide*. Explains the basics of installing and using DI-Guy products.
- *DI-Guy Scenario Users Guide*. Explains how to use DI-Guy Scenario to create scenarios with realistic human activity.
- *DI-Guy Content Reference Manual*. Explains how to customize DI-Guy characters and add new character models. It also describes how to use DI-Guy Expressive Faces.
- *DI-Guy AI Users Guide*. Explains how to use DI-Guy AI to add intelligent behavior to DI-Guy characters.
- *DI-Guy Motion Editor Users Guide*. Explains how to import, edit, and customize DI-Guy motions.

These documents are available from the start menu under DI-Guy, or by browsing to `./doc`.

Try the Help

MAK's HTML5 help (the *DI-Guy Documentation Center*) is the best way to use our documentation. It's easy to navigate, with improved natural language search tools and filters to focus your search. You can click the illustrations to see high-resolution enlargements that show more detail. It's built into the product and installed locally — simply press **F1** to launch the help in your browser. You do not need to be connected to the internet to use it.

 If you attempt to view the help in a browser that does not fully support HTML5, you may experience issues with some features.

Derivative Work

Any DI-Guy content (model, texture, motion, and so on) that you modify or copy is a derivative work that is protected under copyright law by the same rules that apply to content delivered with DI-Guy. This derivative content may be used only on computers that have a valid DI-Guy license. Please contact MAK Technologies to discuss licensing for other uses of derivative models.

MAK Products

The MAK Technologies product line includes:

- **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™.** VR-TheWorld 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 DI-Guy 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 these typographic conventions:

Sans Serif Bold	Indicates a key on the keyboard or menu options. Mentions of letter keys do not require you to press the Shift key. If Shift is necessary, that will be included, for example, Shift+Q .
	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 DI-Guy home directory. For example, the directory *./DI-Guy13.7.1/doc* appears in the text as *./doc*.

DI-Guy Conventions

The DI-Guy documentation uses these conventions used for entity coordinates.

Convention	Indicates
XYZ	X = forward, Y = to the left Z = up
Yaw, Roll, Pitch	Orientation is applied in the order YRP. Yaw = rz – orientation about the vertical axis Roll = rx – orientation about the fore-aft axis Pitch = ry – orientation nose down, nose up

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

DI-Guy 13.7.1 uses the indicated versions of these libraries:

- VR-Link 5.9.1
- 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

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

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

These libraries are covered by the ZLib license:

- Bullet Physics
- LibPNG
- LibSDL
- Minizip
- Zlib

These 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

These libraries are covered by the BSD license:

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

These libraries are covered by the commercial licenses:

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

These 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 1.63 - 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>.

Lua

Some MAK products use the Lua programming language (www.lua.org). Its license is as follows:

Copyright © 1994–2012 Lua.org, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

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 AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Scintilla and SciTE

License for Scintilla and SciTE. Copyright 1998-2002 by Neil Hodgson <neilh@scintilla.org>. All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS

OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.



1. Introduction to DI-Guy

This chapter provides a high-level description of DI-Guy features.

1.1. Introduction	18
1.2. How Is DI-Guy Used?	18
1.2.1 Expressive Faces	19
1.2.2 Vehicle Characters	19
1.3. High-Performance Real-time Motion Engine	19
1.4. DI-Guy Works in any Rendering Environment	20
1.5. Networking with DI-Guy	20

1.1. Introduction

DI-Guy is software for adding realistic human characters to real-time visual simulations. DI-Guy characters move realistically, respond to simple high-level commands, and travel about the environment as directed. DI-Guy characters make seamless transitions from one activity to the next, moving naturally like real people.

Whether you are a seasoned visualization professional or a novice, the DI-Guy SDK will simplify and speed your task of adding realistic lifelike humans to real-time 3D simulators, letting you save time and money.

- DI-Guy automatically creates natural-looking smooth behavior, even when a character changes its behavior from one action to the next.
- You can add characters to a simulation with as few as six high-level function calls, or tweak their behavior and appearance using all 400 function calls available in the DI-Guy API.
- High-level control allows you to concentrate on your application, rather than on the time-consuming details of low-level animation.
- DI-Guy is distributed with easy-to-follow example programs.

1.2. How Is DI-Guy Used?

Here are some of the ways people are using DI-Guy today:

- Ground and urban combat training.
- Mission planning and after-action review.
- Peacekeeper training.
- Law-enforcement training.
- Driving simulators.
- Urban visualization.
- Architectural walk-throughs.
- Disaster evacuation planning.

DI-Guy is used by all branches of the US Armed Forces and by commercial and military organizations worldwide.

DI-Guy comes with everything you need to add lifelike humans to your simulation:

- Photo-realistic human models that are fully textured, articulated, and come with multiple Levels of Detail (LODs).
- A library of human behavior with over 2000 motions, transitions and gestures.
- A high-performance motion engine that lets you command DI-Guy in real time.
- A powerful, easy-to-use C++ API, the DI-Guy API.

DI-Guy includes models and behavior for hundreds of human characters that represent people from all walks of life, including:

- Soldier characters with an extensive array of uniforms, weapons, and behaviors.
- Flight deck crew (LSE, LSO, and so on.)
- Policemen, firemen, and first responders equipped with gas masks and other hazardous (MOPP) gear.
- Suspects and enemy combatants.
- A wide range of civilian men, women, and children from a variety of cultural and ethnic groups.
- Divers, dancers, street sweepers, people using wheelchairs, bicyclists, and travelers.
- Animals, including horses, mules, dogs, cows, kangaroos, and riders on horseback.

DI-Guy also includes hundreds of vehicles and props to support your human performances.

DI-Guy uses industry standard OpenFlight and Collada models and is easy to customize.

1.2.1 Expressive Faces

Expressive Faces is an optional module available for DI-Guy. Expressive Faces are animated 3D heads that can portray different emotional states, blink, and lip-sync to speech. To activate the expressive faces module, you need to initialize and deinitialize the Expressive Faces module and have a valid Expressive Faces license.

DI-Guy Expressive Faces uses FaceFX technology. See the [DI-Guy Content Reference Manual](#) for more information.

1.2.2 Vehicle Characters

Although DI-Guy is most often associated with high-quality human characters, it also includes vehicle character types. Some generic vehicles have limited visual functionality. However, vehicle characters using the `vehicle_complete` type have the following capabilities:

- Rolling and turning wheels.
- Smooth acceleration and deceleration.
- Doors that open and close.
- Body that can rock.
- Aimable turret.

To get these capabilities, use character type "vehicle_09" or other vehicles defined as `vehicle_complete`.

See the [DI-Guy Scenario Users Guide](#) for more information about complete vehicles.

1.3. High-Performance Real-time Motion Engine

An advanced motion engine ensures that DI-Guy entity behavior is lifelike and realistic. DI-Guy comes with over 2,000 motions and transitions. You can modify any of these motions or create and add your own new motions using the optional DI-Guy Motion

Editor. The DI-Guy Motion Editor auditions and exports your new motions directly into your characters' action tables for immediate use in DI-Guy. (See the [DI-Guy Motion Editor Users Guide](#) to learn more.)

DI-Guy is real-time. DI-Guy provides excellent real-time performance through use of optimizations, such as:

- Level-of-detail (LOD) switching.
- Motion level-of-detail (MLOD) switching.
- Character Performance Level (CPL) switching.
- Object-level culling.
- The DI-Guy Load Manager.

1.4. DI-Guy Works in any Rendering Environment

DI-Guy SDK works out of the box with OpenGL, DirectX, OpenSceneGraph (OSG), and Vega Prime. DI-Guy also works with many third-party rendering products including:

- Raydon BARE
- Rockwell Collins EPX
- VR-Vantage
- Quantum3D Mantis
- FlightSafety VITAL X
- Lockheed Martin SEView
- CATI X-IG
- DiamondVisionics

If you want to use DI-Guy in a simulation environment that is not on this list, the DI-Guy Graphics API module is also available to allow you to integrate tightly with whatever rendering environment you use. Please contact support (<https://www.mak.com/support>) for assistance.

1.5. Networking with DI-Guy

DI-Guy supports DIS, HLA1.3, HLA1516, and HLA Evolved networking. The SDK includes functions that simplify the task of getting good human behavior when you integrate DI-Guy into your networked application. We also promote use of the DI-Guy Custom FOM and DI-Guy custom PDUs to get higher fidelity performances in a distributed networking environment than is available with the standard HLA and DIS protocols.

Programming examples are included that show how to integrate DI-Guy into an application that uses VR-Link. VR-Link is not sold with the DI-Guy SDK. You can use the example to help create a networking solution using VR-Link or an alternative networking solution.

You can visualize human entities simulated by OneSAF, DISAF, JSAF, ExportSAF and other related SAFS with IGs, Stealths, and viewers enabled with DI-Guy.



2. Installing DI-Guy

This chapter explains how to install DI-Guy.

2.1. Introduction	22
2.2. Installing DI-Guy on Windows	22
2.3. Installing DI-Guy on Linux	23
2.4. Installing DI-Guy PDB Files for Debugging Purposes	24
2.5. DI-Guy Directory Structure	26
2.6. Installing and Setting Up the MAK License Manager	27
2.7. Troubleshooting Performance and Display Issues	29

2.1. Introduction

To install DI-Guy you install an applications package and a data package. All DI-Guy applications are part of the same installation package. Your access to them is controlled by the license. DI-Guy is provided on DVD or as downloaded executable installers (Windows) or tarred zipped (Linux) files.

2.2. Installing DI-Guy on Windows

To install DI-Guy on Windows:

1. On the DI-Guy DVD or using the installation file you downloaded, run the DI-Guy-Apps installer. Be sure to use the version with the correct compiler and bit level.
2. Follow the instructions of the installation wizard.
3. Run the data installer.
4. Follow the instructions of the installation wizard.

2.2.1 Installing DI-Guy in Silent Mode

You can install DI-Guy applications in a near-silent mode, without an end-user interface. The default installation directory is *C:\MAK\DI-Guy-13.7.0*. The best way to run the silent install is to make sure that the data package is in the same location as the product installer.

► **To install DI-Guy in silent mode:**

1. Install the application using the appropriate command, for example:

```
diguy13.7-win64-vc15-YYYYMMDD.Setup.exe /SILENT
```

This displays the splash screen and then is completely silent while installing the application.

2. If the data package installer is in the same folder as the application installer, it installs after the product finishes installing.

If it is not, or you want to manually install the data, you can extract the *diguy13.7-data-20231121.diguyData.MAK-targz* package into your DI-Guy installation directory.

2.2.2 The DI-Guy Environment Variable

For DI-Guy applications, starting with 13.7, you should only use the DIGUY environment variable if necessary for your configuration. If you have it set for a previous release of DI-Guy, change or remove the variable.

Older Versions

DI-Guy applications of version 13.6 and earlier find data and other files using the DIGUY environment variable. If DI-Guy is installed in *C:\MAK\DI-Guy-13.6.0*, for example, the DIGUY environment variable will be set to *C:\MAK\DI-Guy-13.6.0*.

The DI-Guy SDK Windows installer automatically sets the DIGUY variable to the correct value and adds a DI-Guy folder to the path.

In the standard Windows command processor, %DIGUY% refers to the value of the environment variable DIGUY.

To change the value of this variable, in a command shell, enter:

```
prompt> set DIGUY=c:\Another-DI-guy
```

To change the environment for newly started command shells and for programs that run by clicking Windows Explorer icons, use the Control Panel System applet.

2.2.3 Managing Multiple Versions of DI-Guy on Windows

If for some reason you have multiple versions of DI-Guy prior to 13.7 on your machine, you can do so using the DIGUY environment variable and your PATH environment variable. It is often useful to create a script on a multiple installation system to handle clearing and setting these two environment variables.

► **To install DI-Guy Applications without uninstalling an existing DI-Guy installation:**

1. When the installation wizard prompts:
Previous DI-Guy Applications installation detected. Uninstall?
click No.
2. When the installation wizard prompts:
Leave current installation alone and continue?...
click OK.
3. Choose a folder for installation that will not overwrite an installation you wish to preserve. The wizard displays:
There is already a system environment variable named DIGUY, ...
4. Click OK.

► **To install DI-Guy Data without uninstalling the existing DI-Guy:**

1. When the installation wizard prompts:
Previous DI-Guy Data installation detected. Uninstall?
click No.
2. When the installation wizard prompts:
Leave current installation alone and continue?...
click OK.
3. Install the data in the same folder as the new DI-Guy Applications installation.

2.3. Installing DI-Guy on Linux

Linux versions of the DI-Guy are provided as compressed tar files on DVD, or as downloaded files.

Different versions of DI-Guy are built to run with particular versions of the gcc compiler, often associated with a particular build of Red Hat Enterprise Linux. Consult the [DI-Guy Product Specifications & Dependencies](#) page if you are unsure whether you have the proper version of DI-Guy that corresponds to your target gcc compiler version.

DI-Guy Linux distributions come with an associated README file. Please consult this file for more information about installing the particular version of DI-Guy for Linux.

► **To install DI-Guy on Linux:**

1. Create the directory in which you want to install DI-Guy.
2. Copy the tar file to the install directory.
3. Uncompress and untar the applications file, for example:

```
tar -vxzf diguy-applications-13.6.0-linux_x86-64_gcc44.tar.gz
```

4. Uncompress and untar the data files, for example:

```
tar -vxzf diguy-data-13.6.0-linux.tar.gz
```

2.4. Installing DI-Guy PDB Files for Debugging Purposes

When a MAK ONE application crashes, it can be difficult to determine the cause of the crash. The application generates a *.dmp* file that contains valuable information about the state of the application at the time of the crash. MAK technical support can analyze this file to help diagnose the issue.

In addition, with the application's PDB (program database) files, the application is able to generate a call stack log file (*.callstack.log*) of the application at the time of the crash. The call stack log provides information in plain text about the sequence of functions that were called leading up to the crash, which is also helpful in identifying the specific cause.

By providing the *.dmp* or *.callstack.log* files to technical support, you help support personnel to resolve the issues. They use specialized debugging tools to analyze the files and determine the cause of the crash, and that information aids in the development of a solution to prevent future crashes.

Due to security reasons or organization policies, some customers cannot send the *.dmp* file to technical support. Because the call stack log file is a plain text file, it is easier to inspect and analyze than the *.dmp* file. Users can easily attach it to an email and pass it out of secure areas with visual inspection, making it a convenient option for customers with restrictions on sharing binary files.

It is important to note that the call stack log file alone might not be sufficient for support personnel to fully diagnose the issue. They may require more information to gain a complete understanding of the issue. A detailed procedure for reproducing the crash often helps them solve the problem you encountered.

- Support for the generation of the call stack log file was introduced in MAK ONE 2024. It is supported only in MAK ONE products part of that release and later versions.
- The PDB files are part of the Linux installation. You do not need to install them separately.

2.4.1 Installing PDB Files on Windows

Before installing the PDB files on Windows, you must install DI-Guy. Next, make sure that the PDB files you are installing are from the same build as the application. When you download the installer from the [MAK ONE Product Downloads](#) page, the appropriate PDB installer is also available for download.

► **To install the PDB files:**

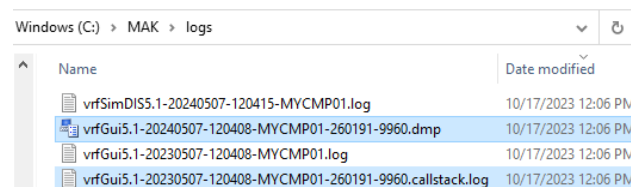
1. After installing DI-Guy, run the PDB installer for that product.
2. The installer looks for the matching product directory by searching the registry and then it automatically sets the install directory to the appropriate path. If that is not correct for your product, see the note below.
3. Complete the installation wizard. The necessary PDB files are installed in the product's `./bin64` directory.

- The default installation directory is `C:\MAK`. See "Default Windows Installation Directory" to learn about setting the default to other locations.
- Be sure that the folder in the Select Destination Location page of the installation wizard is correct. It should be the top-level folder for the product installation.
- If the PDB installer displays a warning that the SVN revisions do not match, make sure that you have the correct PDB installer for the version of DI-Guy that you installed.

2.4.2 Reporting an Issue Using the `.callstack.log` and `.dmp` Files

Send the `.callstack.log` and `.dmp` files (or only the `.callstack.log` if you are unable to provide the `.dmp` file) to MAK support through the Support Portal (<https://www.mak.com/support>) or support@mak.com. The files will have similar names and are located in the central directory for MAK ONE log files (Figure 1). Please also provide information on anything you experienced when the issue occurred.

Figure 1. Example of `.callstack.log` and `.dmp` files for VR-Forces



i The default directory for MAK ONE log files is *C:/MAK/logs*. You can customize the location by specifying a different path in the MAK_LOG_DIR environment variable.

2.5. DI-Guy Directory Structure

Table 1 describes the directories that DI-Guy installs. Table 2 describes the contents of the geometry directory.

Table 1. DI-Guy installation directories

Directory	Contents
3rdparty	Libraries from other vendors that DI-Guy uses. Included is software for multithreading, licensing, glew, networking, sound, physics, and other utility software.
bin64	DI-Guy executables and dynamic libraries.
config	Configuration data describes which character types are available, which appearances are available, and so on. and coordinates the geometry and motion assets of the SDK.
custom	Contents in the custom folder will override certain other files in the original DI-Guy installation. This can include configuration files, geometry, textures, and others.
diguy	Application images.
doc	Documentation.
geometry	3D models, textures, and other data needed to render DI-Guy characters, including OpenFlight and Collada geometry, PNG and RGB textures, and compressed and optimized geometry. Reference shaders are located here as well.
images	Various logos used in documentation and screen shots of most appearances.
include	C++ header files for the DI-Guy API.
lib64	C++ library files for DI-Guy-supplied libraries.
motions	Motions files for animating DI-Guy characters.
programming_examples	Programming examples showing how DI-Guy can be integrated into various graphics environments and how to use various DI-Guy features.
scenarios	Example scenarios created in DI-Guy Scenario.
sfx	Sound files.

Table 2. geometry directory

Directory	Contents
compressed_texture_cache	Textures that have been compressed for better load times and smaller memory footprint.
dae	Files in DAE (COLLADA) format.
facefx	Files needed for the Expressive Faces module.
flt	Geometry files in the OpenFlight format.
fonts	Files used in DI-Guy Scenario for rendering text in OpenGL.
max	Files in 3DS Max format.
maya	Files in Autodesk Maya format.
optimized_geometry_cache	Geometry files that are in a DI-Guy graphics pipeline optimized format.
png	Texture files in the PNG format.
rgb	Texture files in the RGB and RGBA formats.
sets	Terrains that can be used in DI-Guy Scenario.
shaders	Reference geometry shader programs.
tif	Files in TIFF format.

2.6. 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.

 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.

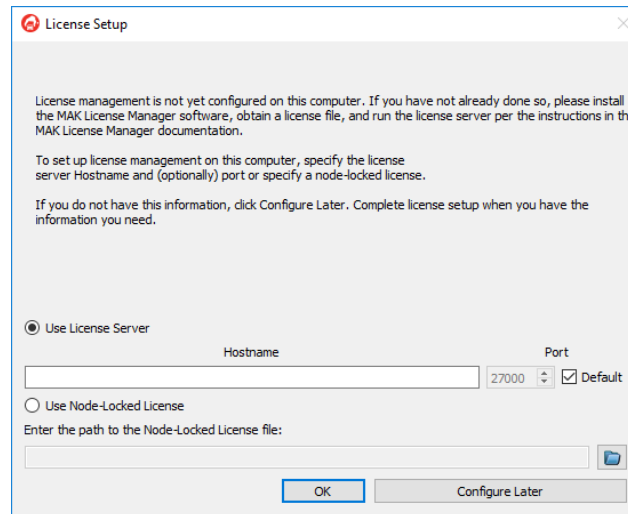
The License Manager installer is included with MAK installation media. It is separate from the product installers. You can download the installers from the [MAK ONE Product Downloads](https://www.mak.com/mak-one-downloads) page at <https://www.mak.com/mak-one-downloads>.

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*.

2.6.1 Specifying the License Server

The first time you run a MAK ONE product on a particular computer, the License Setup dialog box opens (Figure 2). It prompts you to enter the hostname of the license server and, optionally, a port number.

Figure 2. 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, enter `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.

2.7. Troubleshooting Performance and Display Issues

There are a few issues that require you to add the application name to the *BlackApps.dat*:

- Nahimic can severely impact performance and frame rates because it interacts with certain hardware capabilities, vantage leverages, in an extremely detrimental way.
- Under certain circumstances, opening DI-Guy on a second display prevents the program from starting.
- In addition, launching from the main display and then moving the mouse cursor to the other monitor causes the program to freeze and not respond.
- Nahimic can also sometimes cause DI-Guy to crash.

If you experience any of these issues, edit the Nahimic *BlackApps.dat* file and add the application name (for example, DIGuyCharacterViewer.exe, DIGuyMotionEditor.exe, or DIGuyScenario.exe) to the list. You can search *C:\ProgramData* for *BlackApps.dat* to locate it. This file might be found in *C:\ProgramData\A-Volute\A-Volute.Nahimic\Modules\Scheduled\Configurator*, or *C:\ProgramData\A-Volute\A-Volute.28054DF1F58B4\Modules\ScheduledModules\Configurator*, or something similar.



Index

C

- callstack.log file, 24
- crash, 29
- crash, application, 25

D

- debugging
 - callstack.log file, 24
 - dmp file, 24
 - file location, 25
- DIGUY
 - environment variable, 22-23
- DI-Guy
 - directory structure, 26
 - installing, 22
- directory
 - DI-Guy, 26
 - structure, 26
- DIS, 20
- dmp file
 - debugging, 24
 - location, 25
- documentation
 - help, 6

E

- environment variable
 - DIGUY, 22-23
 - MAKLMGRD_LICENSE_FILE, 28

- PATH, 23
- Expressive Faces, 19

F

- FaceFX, 19
- FLEXlm, 27

G

- geometry
 - directory, 26

H

- help
 - HTML5, 6
- High Level Architecture, See HLA
- HLA
 - 1.3, 20
 - 1516, 20
 - Evolved, 20
- hostname
 - license server, 27

I

- installing
 - on Linux, 23
 - on Windows, 22

L

license manager, 27

license server
hostname, 27

Linux
installing DI-Guy, 23

log file
callstack, 24

M

MAKLMGRD_LICENSE_FILE environment variable,
28

Motion Editor, 19

P

PATH environment variable, 23

PDB files
Linux, 25
Windows, 25

performance
optimizing, 29

T

troubleshooting, 29

V

variable, environment, 28



MAK ONE

DIG-13.7.1-10-240819