DI-Guy Getting Started

Users Guide

DI-Guy Getting Started

Users Guide

Contents

Index

Preface

This manual is written for persons who want to create head animations for use with the Expressive Faces module for DI-Guy. 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. It also assumes that you are familiar with DI-Guy Scenario.

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

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.* Contains tables of contents for all DI-Guy manuals and a master index to help you quickly locate the correct manual for the information you need.
- *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*.

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 VT MÄK to discuss licensing for other uses of derivative models.

MÄK Products

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

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

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

- MÄK RTI. An RTI (Run-Time Infrastructure) is required to run applications using the High Level Architecture (HLA). The MÄK 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 a 2D and 3D views of a simulated environment.

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

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

- VR-Vantage™. VR-Vantage is a line of products designed to meet your simulation visualization needs. It includes three end-user applications (VR-Vantage Stealth, VR-Vantage PVD, and VR-Vantage IG) and the VR-Vantage Toolkit.

  – VR-Vantage Stealth displays a realistic, 3D view of your virtual world, a 2D plan view, and an exaggerated reality (XR) view. Together these views provide both situational awareness and the big picture of the simulated world. You can move your viewpoint to any location in the 3D world and can attach it to simulation objects so that it moves as they do.

  – VR-Vantage IG is a configurable desktop image generator (IG) for out the window (OTW) scenes and remote camera views. It has most of the features of the Stealth, but is optimized for its IG function.

  – VR-Vantage PVD provides a 2D plan view display. It gives you the big picture of the simulated world.

  – SensorFX. SensorFX is an enhanced version of VR-Vantage that uses physics based sensors to view terrain and simulation object models that have been materially classified. It is built in partnership with JRM Technologies.

  – The VR-Vantage Toolkit is a 3D visual application development toolkit. Use it to customize or extend MÄK'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.

- MÄK 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™. The DI-Guy product line is a set of software tools for real-time human visualization, simulation, and artificial intelligence. Every DI-Guy software offering comes with thousands of ready-to-use characters, appearances, and motions. DI-Guy enables the easy creation of crowds and individuals who are terrain aware, autonomous, and react intelligently to ongoing events. Save time, money and create outstanding simulations with DI-Guy. The DI-Guy product line includes the following products:

  – The DI-Guy SDK. Embed the DI-Guy library in your real-time application and populate your world with lifelike human characters.

  – DI-Guy Scenario™. Author and visualize human performances in a rich, user-friendly graphical environment. Use DI-Guy Scenario as an end visualization application or save scenarios and load them into your DI-Guy SDK enabled application.

  – ECOSim. Enhanced Company Operations Simulation (ECOSim) is a company-level training simulation that teaches leaders how best to deploy troops, UAVs, convoys, and other assets. ECOSim focuses on ease-of-use, rapid scenario generation, runtime operator control, and realistic and reactive human simulation.

  – DI-Guy AI. Generate crowds of autonomous characters to quickly populate your worlds with hundreds and thousands of terrain-aware, collision avoiding DI-Guys. Used as a module on top of DI-Guy Scenario and DI-Guy SDK.

  – Expressive Faces Module. Enable DI-Guy characters to have faces that display emotion, eyes that look in directions and blink, and lips that sync to sound files.

  – DI-Guy Motion Editor. Create or customize motions to your particular needs in an easy-to-use graphical application.

- 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.

- WebLVC Server. WebLVC Server implements the server side of the WebLVC protocol so that web-based simulation federates can participate in a distributed simulation. It is part of the WebLVC Suite, which includes the server and several sample applications that work with VR-Forces and VR-Vantage.

How to Contact Us

For DI-Guy technical support, information about upgrades, and information about other MÄK products, you can contact us in the following ways:

Telephone

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

E-mail

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

Internet

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

Post

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

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

Document Conventions

This manual uses the following typographic conventions:

`Monospaced`	Indicates commands or values you enter.
`Monospaced Bold`	Indicates a key on the keyboard.
`Monospaced Italic`	Indicates command variables that you replace with appropriate values.
Blue text	A hypertext link to another location in this manual or another manual in the documentation set.
{ }	Indicates required arguments.
[ ]	Indicates optional arguments.
|	Separates options in a command where only one option may be chosen at a time.
(|)	In command syntax, indicates equivalent alternatives for a command-line option, for example, (`-h` | `--help`).
/	Indicates a directory. Since MÄK products run on both Linux and Windows PC platforms, we use the / (slash) for generic discussions of pathnames. If you are running on a PC, substitute a \ (backslash) when you type pathnames.
Italic	Indicates a file name, pathname, or a class name.
sans Serif	Indicates a parameter or argument.
➤	Indicates a one-step procedure.
Menu → Option	Indicates a menu choice. For example, an instruction to select the Save option from the File menu appears as: Choose **File → Save**.
	Click the icon to run a tutorial video in the default browser.
i	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 *vrforces13.2/doc* appears in the text as *./doc.*

DI-Guy Conventions

The following table lists the conventions used for entity coordinates in DI-Guy documentation.

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

MÄK software products may use code from third parties. This section contains the license documentation required by these third parties.

Boost License

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

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

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

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

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

libXML and libICONV

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

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

Lua

Some MÄK 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 DEAL-INGS IN THE SOFTWARE.

1. Introduction to DI-Guy

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

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.

1.1.1. 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.

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. Please see *DI-Guy Content Reference Manual* for more information.

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.

Please see *DI-Guy Scenario Users Guide* for more information about complete vehicles.

1.1.2. 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 will audition and export your new motions directly into your characters' action tables for immediate use in DI-Guy.

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.

1.1.3. 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@mak.com for assistance.

1.1.4. 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 and set the DIGUY environment variable.

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. Managing Multiple Versions of DI-Guy on Windows

If for some reason you need to have multiple versions of DI-Guy 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. Unin-
     stall?
   ```

   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 will display:

   ```
   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 intallation.

2.2.2. The DI-Guy Environment Variable

DI-Guy applications find data and other files by looking up the environment variable DIGUY. If DI-Guy is installed in *C:\DI-Guy-13.1*, for example, the DIGUY environment variable will be set to C:\DI-Guy-13.0.

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.

2.3. Installing DI-Guy on Linux

Linux versions of DI-Guy are provided as compressed tar files on DVD, or as down-loaded 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 versions page http://www.mak.com/support/product-versions/humans/di-guy if you are unsure if 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-13.2.0-linux_x86-64_g-
      cc44.tar.gz
   ```

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

   ```
   tar -vxzf diguy-data-cd1-13.2.0-linux.tar.gz
   tar -vxzf diguy-data-cd2-13.2.0-linux.tar.gz
   ```

2.4. DI-Guy Directory Structure

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

Table 2-1: 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.
bin	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.
flexlm	Files needed for DI-Guy licensing using FlexLM license software.
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.
lib	C++ library files for DI-Guy-supplied libraries.
me	Runtime data for the Motion Editor.
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 2-2: 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.

2.5. Installing and Setting Up the MÄK License Manager

Before you can use a MÄK product, you must obtain a valid license file, install the MÄK 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 MÄK 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 MÄK product, you do not have to install it again. You just need to make sure you have licenses for your newly installed products.

The License Manager installer is included on MÄK installation media. It is separate from the product installers. You can download the installers from our web site at http://www.mak.com/license-support.html or you can download directly from:

- Windows:
  ftp://ftp.mak.com/out/MAKLicenseManager-win-setup.exe (for 32-bit systems)
  ftp://ftp.mak.com/out/MAKLicenseManager-win64-setup.exe
- Linux:
  ftp://ftp.mak.com/out/MAKLicenseManager-linux-setup.tar.gz (for 32-bit systems)
  ftp://ftp.mak.com/out/MAKLicenseManager-linux64-setup.tar.gz

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

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

2.5.1. Specifying the License Server

The first time you run a MÄK application on a particular computer, the License Setup dialog box opens (Figure 2-1). It prompts you to enter the hostname of the license server and optionally, a port number.
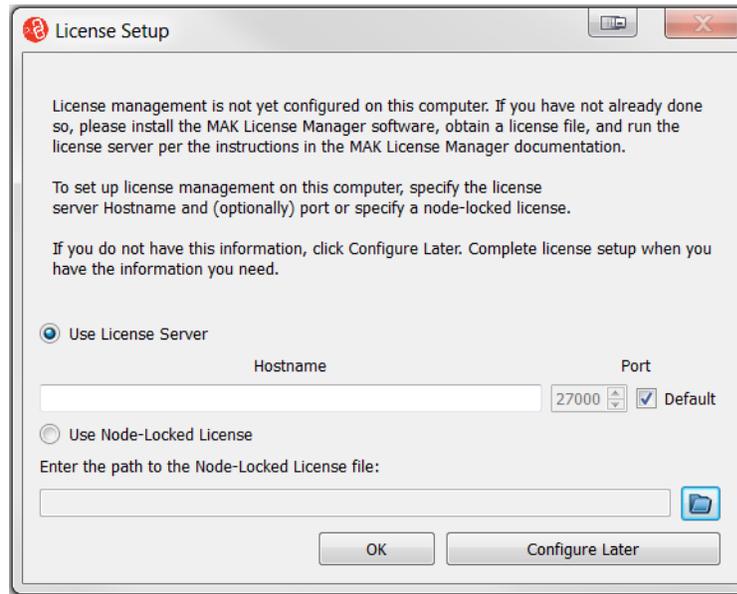

Figure 2-1. License Setup dialog box

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 MÄK applications until you set up license management.

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

---

i

- If you are running MÄK 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 MÄK application.
- You can clear the saved license configuration by deleting the cache file. On Windows XP, it is *C:\Documents and Settings\user_name\Application Data\MAK\license<n>.xml*. On Windows 7, it is *C:\Users\user_name\AppData\Roaming\MAK\licenses<n>.xml*. (The *AppData* directory is hidden by default.) On Linux, it is *.mak/license<n>.xml*. (There may be more than one cache file, for example, *licenses1.xml* and *licenses2.xml*.)

The MAKLMGRD_LICENSE_FILE Environment Variable

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

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

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

Windows

To add the MAKLMGRD_LICENSE_FILE in Windows:

1. Open the Control Panel.

2. Click System and Security.

3. Click System.

4. In the sidebar menu, click the Advanced System Settings. The System Properties dialog box opens.

5. Click Environment Variables. The Environment Variables dialog box opens.

6. Click New. The New System Variable dialog box opens.

7. In the Variable Name field, enter MAKLMGRD_LICENSE_FILE.

8. In the Variable Value field, enter `@server_name`, where `server_name` is the name of the license server.

9. Click OK to back out of each dialog box and set the variable.

i   Different versions of Windows have slightly different wording of the various Control Panel options.

Linux

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

```
setenv MAKLMGRD_LICENSE_FILE @oak
```

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

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

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

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

Index

D

dialog box
    License Setup 2-8
DIGUY
    environment variable 2-2, 2-3
DI-Guy
    directory structure 2-5
    installing 2-2
directory
    DI-Guy 2-5
directory structure 2-5
DIS 1-5

E

environment variable
    DIGUY 2-2, 2-3
    MAKLMGRD_LICENSE_FILE 2-9
    PATH 2-2
Expressive Faces 1-3

F

FaceFX 1-3
FLEXlm 2-7

G

geometry
    directory 2-5

H

HLA 1.3 1-5

HLA 1516 1-5
HLA Evolved 1-5
hostname
    license server 2-8

I

installing
    on Linux 2-4
    on Windows 2-2

L

License Manager 2-7
license server
    hostname 2-8
License Setup dialog box 2-8
Linux
    installing DI-Guy 2-4

M

MAKLMGRD_LICENSE_FILE
    environment variable 2-9
Motion Editor 1-4

P

PATH
    environment variable 2-2

Link - Simulate - Visualize