Informix Dynamic 4GL

Installation and Configuration Guide

Version 3.01
June 2000
Part No. 000-6754

Published by Informix® Press

Informix Corporation
4100 Bohannon Drive
Menlo Park, CA 94025-1032

Documentation Team: Twila Booth, Elaina Van Haas, Jennifer Leland

Table of Contents

Introduction

In This Introduction 3
About This Guide 3
- Organization of This Guide 3
- Types of Users 4
- Software Dependencies 4
- Compiling to C Code 4
- Assumptions About Your Locale 5

Documentation Conventions 5
- Typographical Conventions 6
- Icon Conventions 6

Additional Documentation 8
Informix Welcomes Your Comments 9

Chapter 1 Dynamic 4GL Architecture and Installation Overview

In This Chapter 1-3
Dynamic 4GL Architecture 1-3
- Architectural Overview 1-4
- Definition of Terms 1-6

Overview of the Installation Process 1-8

Chapter 2 Installing Informix Dynamic 4GL

In This Chapter 2-3
Before Installing Dynamic 4GL 2-3
Upgrading from a Previous Version 2-5
Application Server Requirements 2-6
- Supported Operating Systems 2-8
- C Compiler Requirements 2-9
- Informix Client SDK and ESQL/C Requirements 2-10

Installing Dynamic 4GL on UNIX 2-11

Installing Dynamic 4GL on Windows NT 2-14
Testing Your Database Connection on Windows NT 2-18
Dynamic 4GL Directories 2-19
Licensing 2-20
Working with Licenses on UNIX Systems 2-20
Reactivating a License 2-21
Workarounds for Licensing Errors 2-22
Working with Licenses in Windows NT 2-23

Chapter 3 Installing the Windows Client

In This Chapter 3-3
Windows Client Requirements 3-4
Front-End Client Requirements 3-4
Application Server Requirements 3-5
Installing the Windows Client 3-5
Installing the Windows Client on a Network 3-6
Configuring the Windows Client 3-7

Chapter 4 Installing the X11 Client

In This Chapter 4-3
X11 Client Requirements 4-4
Front-End Client Requirements 4-4
Application Server Requirements 4-4
Installing the X11 Client 4-5
Configuring the X11 Client 4-7

Chapter 5 Installing the Java Client

In This Chapter 5-3
Before Installing the Java Client 5-4
Location of Java Client Installation Components 5-4
Aliases 5-4
Pathname Variables 5-4
Java Client Requirements 5-5
Front-End Client Requirements 5-5
Web Browser Requirements 5-6
Web Server Requirements 5-7
Application Server Requirements 5-8
Installing the Swing Components of the Java Client 5-8
Installing the Java Client on UNIX 5-9
Configuring the Java Client on UNIX 5-11

Installing the Java Client on Windows NT 5-12
Configuring the Java Client on Windows NT 5-13
Verifying CLASSPATH 5-14
Verifying the CLASSPATH Setting 5-14
Setting CLASSPATH 5-15
Configuring the Servlet Engine 5-17
Servlet Mapping 5-17
Servlet Aliases 5-17
Apache with Apache Jserv on UNIX 5-19
Netscape with JRun on UNIX 5-19
Sun Java Web Server on Windows NT 5-21

Chapter 6 Installing the HTML Client

In This Chapter 6-3
Before Installing the HTML Client 6-4
HTML Client Requirements 6-4
Front-End Client Requirements 6-4
Web Server Requirements 6-4
Application Server Requirements 6-5
Installing the HTML Client on UNIX 6-5
Installing the HTML Client on Windows NT 6-7
Manually Installing HTML Client Components 6-8
Installing the HTML Client on the Web Server 6-9
Installing the HTTP Server on the Application Server . . . 6-10
Installing the HTML Documentation on the Web Server . . . 6-11
Installing the Example 6-12
Configuring the HTML Client on UNIX and Windows NT . . 6-12

Index

Introduction

In This Introduction 3

About This Guide 3
- Organization of This Guide 3
- Types of Users 4
- Software Dependencies 4
- Compiling to C Code 4
- Assumptions About Your Locale 5

Documentation Conventions 5
- Typographical Conventions 6
- Icon Conventions 6
  - Comment Icons 7
  - Feature, Product, and Platform Icons 7

Additional Documentation 8

Informix Welcomes Your Comments 9

In This Introduction

This introduction provides an overview of the information in this guide and describes the conventions it uses.

About This Guide

The *Informix Dynamic 4GL Installation and Configuration Guide* provides you with the conceptual and task information you need to install and configure the Informix Dynamic 4GL development and runtime packages, related files, and clients.

Organization of This Guide

This guide includes the following chapters:

- Chapter 1, "Dynamic 4GL Architecture and Installation Overview," describes the Informix Dynamic 4GL architecture for installation and development and provides an overview of the installation process.
- Chapter 2, "Installing Informix Dynamic 4GL," explains how to install the Dynamic 4GL development and runtime software on both UNIX and Windows NT.
- Chapter 3, "Installing the Windows Client," explains how to install the Windows Client, an optional software component that enables graphical display of Dynamic 4GL applications, on Windows 95, 98, and NT front-end clients.
- Chapter 4, "Installing the X11 Client," explains how to install the Windows Xll Client, an optional software component that enables graphical display of Dynamic 4GL applications on UNIX front-end clients running the X Window System.

- Chapter 5, "Installing the Java Client," explains how to install the Java Client, an optional software component that enables a front-end client to connect to your Dynamic 4GL application from a Java-enabled Web browser. You do not need to write Java code to use the Java Client.
- Chapter 6, "Installing the HTML Client," explains how to install the HTML Client, an optional software component that enables a front-end client to connect to your Dynamic 4GL application through a Web browser.

For help with new terms, see the glossary in the *Informix Dynamic 4GL User Guide*.

Types of Users

This guide is written for system administrators and application developers who need to install the Dynamic 4GL compiler, runtime, related files, or clients.

Software Dependencies

This guide is written with the assumption that you are using an Informix database server, and that you have installed and correctly configured either the Informix Client Software Developer's Kit or ESQL/C. Dynamic 4GL uses the ESQL/C libraries you have installed with your database server to create the runtime for your application. Availability of Dynamic 4GL features is dependent upon the version of the database server and ESQL/C that you have installed.

Compiling to C Code

Dynamic 4GL features introduced in Version 3.0 do not support compilation to C code. If you are not using these newer features, compilation to C code is still possible; however, you are advised to use p-code.

Assumptions About Your Locale

Informix products can support many languages, cultures, and code sets. All the information related to character set, collation and representation of numeric data, currency, date, and time is brought together in a single environment, called a Global Language Support (GLS) locale.

This manual is written with the assumption that you are using the default locale, **en_us.8859-1**. This locale supports U.S. English format conventions for dates, times, and currency. In addition, this locale supports the ISO 8859-1 code set, which includes the ASCII code set plus many 8-bit characters such as é, è, and ñ.

If you plan to use nondefault characters in your data or your SQL identifiers, or if you want to conform to the nondefault collation rules of character data, you need to specify the appropriate nondefault locale.

For instructions on how to specify a nondefault locale, additional syntax, and other considerations related to GLS locales, see the *Informix Guide to GLS Functionality*.

Documentation Conventions

This section describes the conventions that this guide uses. These conventions make it easier to gather information from this and other volumes in the documentation set.

The following conventions are covered:

- Typographical conventions
- Icon conventions

Typographical Conventions

This guide uses the following conventions to introduce new terms, illustrate screen displays, describe command syntax, and so forth.

Convention	Meaning
KEYWORD	All primary elements in a programming language statement (keywords) appear in uppercase letters in a serif font.
italics *italics* *italics*	Within text, new terms and emphasized words appear in italics. Within syntax and code examples, variable values that you are to specify appear in italics.
boldface ***boldface***	Names of program entities (such as classes, events, and tables), environment variables, file and pathnames, and interface elements (such as icons, menu items, and buttons) appear in boldface.
monospace *monospace*	Information that the product displays and information that you enter appear in a monospace typeface.
KEYSTROKE	Keys that you are to press appear in uppercase letters in a sans serif font.
→	This symbol indicates a menu item. For example, "Choose **Tools→Options**" means choose the **Options** item from the **Tools** menu.

Tip: *When you are instructed to "enter" characters or to "execute" a command, immediately press* RETURN *after the entry. When you are instructed to "type" the text or to "press" other keys, no* RETURN *is required.*

Icon Conventions

Throughout the documentation, you will find text that is identified by several different types of icons. This section describes these icons.

Comment Icons

Comment icons identify three types of information, as the following table describes. This information always appears in italics.

Icon	Label	Description
	***Warning:***	Identifies paragraphs that contain vital instructions, cautions, or critical information
	***Important:***	Identifies paragraphs that contain significant information about the feature or operation that is being described
	***Tip:***	Identifies paragraphs that offer additional details or shortcuts for the functionality that is being described

Feature, Product, and Platform Icons

Feature, product, and platform icons identify paragraphs that contain feature-specific, product-specific, or platform-specific information.

Icon	Description
GLS	Identifies information that relates to the Informix Global Language Support (GLS) feature
UNIX	Identifies information that is specific to UNIX
WIN 3.1	Identifies information that is specific to Windows 3.1
Windows	Identifies information that is specific to Windows 95, Windows 98, and Windows NT
WIN NT	Identifies information that is specific to Windows NT

These icons can apply to an entire section or to one or more paragraphs within a section. If an icon appears next to a section heading, the information that applies to the indicated feature, product, or platform ends at the next heading at the same or higher level. A ♦ symbol indicates the end of feature-, product-, or platform-specific information that appears in one or more paragraphs within a section.

Additional Documentation

Dynamic 4GL documentation is provided in a variety of formats:

- **Online manuals.** You can print chapters or entire books and do full-text searches for information in specific books or throughout the documentation set. Online manuals are available through Answers OnLine. You can order Answers OnLine on a CD, or if you have access to the Web, visit the following URL: www.informix.com/answers.
- **Printed manuals.** The *Informix Dynamic 4GL User Guide* and the *Informix Dynamic 4GL Installation and Configuration Guide* are available as printed manuals. To order printed manuals, call 1-800-331-1763 or send e-mail to moreinfo@informix.com. Provide the following information when you place your order:
  - The documentation that you need
  - The quantity that you need
  - Your name, address, and telephone number
- **Release notes.** Release notes are located in the **/release** directory where the product is installed. Please examine these files because they contain vital information about application and performance issues.
- **Documentation notes.** Documentation notes are located in the **/release** directory where the product is installed. These notes contain additions and corrections to the manuals.

- **HTML files.** Some additional documentation about Web server configuration is provided in supplementary HTML files. For more information, see "Installing the HTML Documentation on the Web Server" on page 6-11.
- **Related Reading.** For information about developing 4GL applications, refer to your INFORMIX-4GL documentation.

Informix Welcomes Your Comments

Let us know what you like or dislike about our manuals. To help us with future versions of our manuals, we want to know about any corrections or clarifications that you would find useful. Include the following information:

- The name and version of the guide that you are using
- Any comments that you have about the guide
- Your name, address, and phone number

Send electronic mail to us at the following address:

doc@informix.com

This address is reserved for reporting errors and omissions in our documentation. For immediate help with a technical problem, contact Informix Customer Services.

We appreciate your suggestions.

Chapter

1

Dynamic 4GL Architecture and Installation Overview

Dynamic 4GL Architecture 1-3
Architectural Overview 1-4
Definition of Terms 1-6

Overview of the Installation Process 1-8

In This Chapter

This chapter describes the Informix Dynamic 4GL components and the client/server architecture for installation and development, and provides an overview of the installation process.

Dynamic 4GL Architecture

Dynamic 4GL allows a multitier client/server architecture that consists of a:

- **Database server.** Contains your database.
- **Application server.** Contains your 4GL source code, Dynamic 4GL compiler, and the Dynamic Virtual Machine (DVM).
- **Web server.** You need a Web server if you are developing an application for the Web.
- **Front-end client.** Displays your application.

Architectural Overview

The architecture you choose depends on the method by which you choose to display your application. Four front-end display clients are available with Dynamic 4GL:

- **X11 Client.** Allows for graphical display of your compiled application on a UNIX system running the X Window System.
- **Windows Client.** Allows for graphical display of your compiled application on a PC running Microsoft Windows.
- **HTML Client.** Allows for Web deployment of your compiled application on either UNIX or a PC (or MacIntosh) through a standard Web browser.
- **Java Client.** Allows for Web deployment of your compiled application on either UNIX or a PC (or MacIntosh) through a Java-enabled Web browser. (No Java coding is necessary for this display method.)

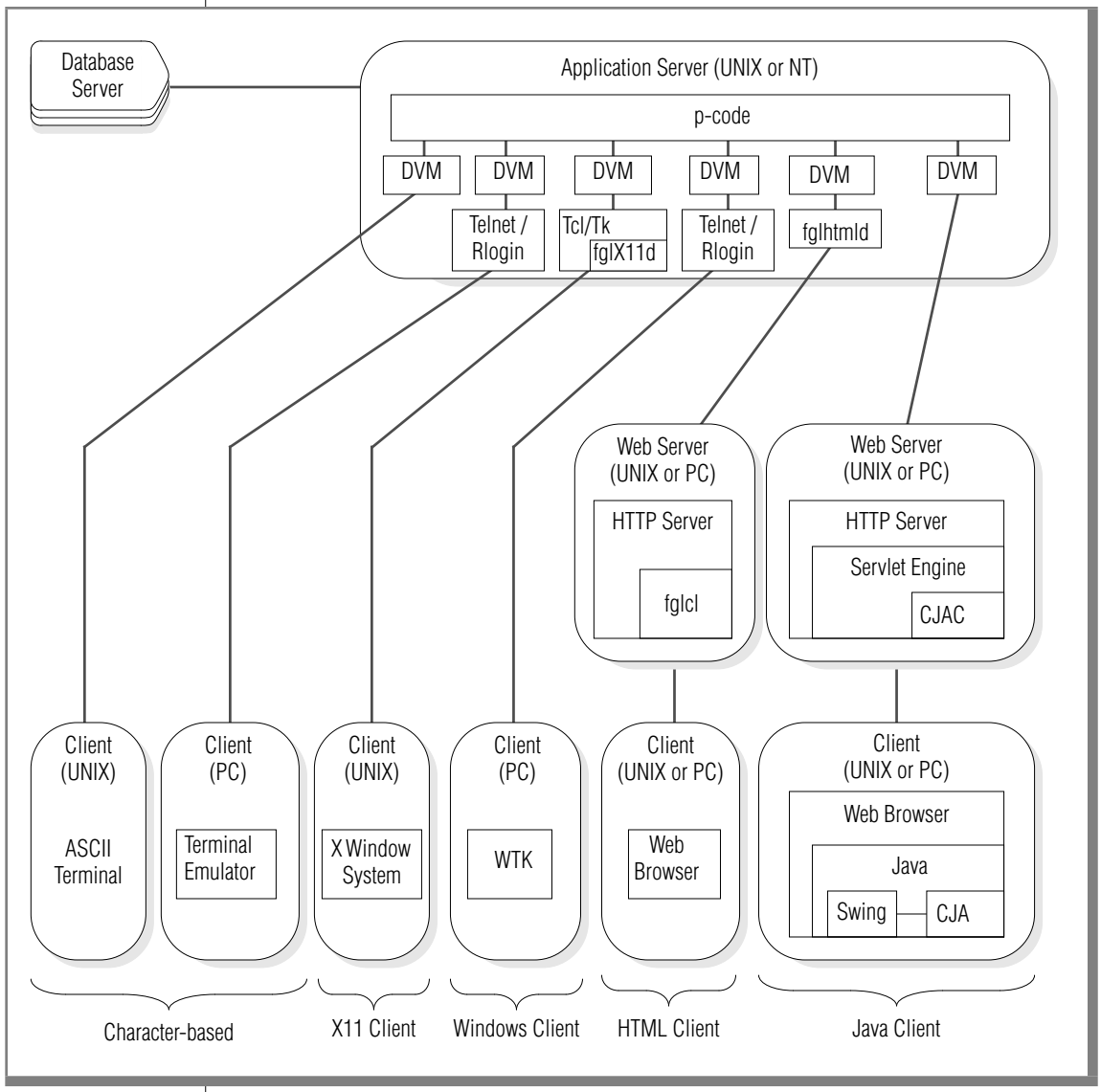
You can also continue to use a character-based display of your compiled application.

All communication from the DVM to the front-end client occurs through TCP/IP. (For Microsoft Windows systems, TCP/IP must be installed in addition to the existing NetBEUI.) This makes it possible to collapse the architecture so that the database server, application server, and Web server can reside on the same computer. This eases the development process. Before you install Dynamic 4GL, you need to determine the number of tiers you want to use for development.

When you deploy your application, you need to expand the architecture in order to build in appropriate security. Also, you need to install various components to your customer's front-end client in order to deploy your application. And, if you are migrating between UNIX and Windows NT during deployment, you need to consider additional factors. For additional information, see the *Informix Dynamic 4GL User Guide*.

Figure 1-1 shows the expanded architecture and location of each component for Dynamic 4GL.

Figure 1-1
Dynamic 4GL Architecture



Definition of Terms

The following terms define the elements of this architecture:

- **Database Server**. Location of Informix database. The database server can be on either UNIX or Windows NT.
- **Application Server**. Location of your compiled Dynamic 4GL application and the runner that executes the application. Typically, the application server is where your 4GL source files and Dynamic 4GL software are installed. The application server can be on either UNIX or Windows NT. The application server and the database server can reside on the same computer.
- **DVM.** The Dynamic Virtual Machine. The DVM is a particular instance of a Dynamic 4GL application at runtime.
- **Web Server**. Location of the Web server component for the HTML Client or the Java Client. The Web Server must be installed and running before you can install the HTML Client or Java Client. If you plan to use the Java Client, the Web server must support Java servlets and be JSDK 2.0 compliant. The Web server can reside on the same computer as the application server.
- **Web Browser**. Communicates with the Web server enabling display of the HTML Client or Java Client. The Web browser must support HTML, Version 3.2 or higher, if you are using the HTML Client. If you are using the Java Client, the Web browser must be Java-enabled (alternatively, you can use a Java appletviewer).
- **X11 Client**. Optional component that enables graphical display of Dynamic 4GL applications on UNIX systems running the X Window System. The X11 Client consists of the Tcl/Tk interpreter and the X11 daemon called **fglX11d**. The X11 daemon is installed automatically on the application server when you perform an automatic installation of Dynamic 4GL on UNIX.

  Alternatively, you can manually install the X11 daemon on another computer with the Tcl/Tk interpreter. This adds an additional tier to your architecture. However, this structure is not advised as it requires an additional installation of the compiler on the computer where the X11 daemon is located. This secondary compiler must be unlicensed.

- **Windows Client**. Optional component that enables graphical display of Dynamic 4GL applications on Microsoft Windows 3.1 (with the Win32s extender), 95, 98, and NT systems. The Windows Client consists of a graphical daemon that is run by a graphical display server named 4GL Server. You must have the Microsoft TCP/IP stack installed before you install the Windows Client. If you are planning to use Windows NT as your application server, you are advised to install the Windows Client before you install the basic Dynamic 4GL software.
- **HTML Client**. Optional component that enables deployment of your Dynamic 4GL application on the Web. The application is displayed in a Web browser that supports HTML, Version 3.2 or higher. The HTML Client consists of a CGI executable called **fglcl** and an HTTP server program called **fglhtmld**. The CGI executable **fglcl** must be installed on the same computer as the Web server. The **fglhtmld** daemon must be installed on the application server (**fglhtmld** works with the application runner that is located on the application server). The Web server and Web browser must be installed, and the Web server configured before you install the HTML Client. The Web server should be running during installation of the HTML Client.
- **Java Client**. Optional component that enables deployment of your Dynamic 4GL application on the Web. The application is displayed in a Java-enabled Web browser (or Java appletviewer). The Web server must support Java servlets and be JSDK 2.0 compliant. The Web server and Web browser must be installed, and the Web server configured before you begin installing the Java Client. The Web server should be running during installation of the Java Client. You then install and configure the CJAC servlets on the Web server, and then install the Swing class libraries in the Web browser on the front-end client. You do not need to write Java code to use the Java Client.

Overview of the Installation Process

The following steps outline the installation process. Read the chapters that follow for specific instructions.

1. Determine the architecture you will use for development.
2. Review the system requirements for installing the compiler and runtime packages and verify that your application server meets all requirements.

   For details, see "Application Server Requirements" on page 2-6.
3. If you are upgrading from a previous version of Dynamic 4GL, you need to take certain steps before you proceed.

   For details, see "Upgrading from a Previous Version" on page 2-5.
4. Start the database server.

   This allows you to test connections between the application server and the database server. It is also required by the installation script in order to create the default p-code runner.
5. Verify that ESQL/C is correctly installed and configured.

   To check the version and settings of ESQL/C on UNIX, enter the command:

   ```
   esql-V
   ```

   To check the ESQL/C settings on Windows NT, attempt to connect to your database through I-Login.

   If ESQL/C is not properly configured, or you are unable to connect to the database, set **INFORMIXDIR**, **INFORMIXSERVER**, and **INFORMIXSQLHSTS**. For details, refer to your ESQL/C or Informix Client Software Developer's Kit documentation.
6. Install the compiler and/or runtime packages following the instructions in Chapter 2, "Installing Informix Dynamic 4GL."
7. Review the system requirements for the front-end client you plan to install and verify that your application server, Web server, and front-end client meet all requirements.

8. If you plan to install one of the Web clients, install and configure your Web server before you proceed.

   Follow the instructions provided in your Web server documentation. Be sure to test the installation and configuration of your Web server before you install the Web client. When you install the Web client, Dynamic 4GL installs additional components on your Web server.

9. Install the front-end client following the instructions in the appropriate chapter:

   - Chapter 3, "Installing the Windows Client"
   - Chapter 4, "Installing the X11 Client"
   - Chapter 5, "Installing the Java Client"
   - Chapter 6, "Installing the HTML Client"

   You can install one or more clients in any order.

Tip: *If you encounter any difficulties with licensing, refer to**"Licensing" on page 2-20**.*

Chapter

2

Installing Informix Dynamic 4GL

Before Installing Dynamic 4GL 2-3

Upgrading from a Previous Version 2-5

Application Server Requirements 2-6
- Supported Operating Systems. 2-8
- C Compiler Requirements 2-9
- Informix Client SDK and ESQL/C Requirements 2-10

Installing Dynamic 4GL on UNIX 2-11

Installing Dynamic 4GL on Windows NT 2-14
- Testing Your Database Connection on Windows NT 2-18

Dynamic 4GL Directories. 2-19

Licensing 2-20
- Working with Licenses on UNIX Systems. 2-20
- Reactivating a License 2-21
- Workarounds for Licensing Errors 2-22
- Working with Licenses in Windows NT 2-23
  - Using the License Manager 2-23
  - Notes on Licensing with Windows NT 2-24

In This Chapter

This chapter tells you how to install the Dynamic 4GL development and runtime software. It includes the following sections:

- Before installing Dynamic 4GL
- Upgrading from a previous version
- Application server requirements
- Installing Dynamic 4GL on UNIX
- Installing Dynamic 4GL on Windows NT
- Dynamic 4GL directories
- Licensing

Before Installing Dynamic 4GL

Before you begin to install Dynamic 4GL, review the following list:

- **Dynamic 4GL Directory (FGLDIR).** Install Dynamic 4GL in its own directory (separate from **Informix Client Software Developer's Kit** and not in **INFORMIXDIR**). You will find this makes upgrading and maintaining Dynamic 4GL easier.
- **Informix Database Server.** Run the database server while you install Dynamic 4GL. This enables you to check the success of the installation by testing the connection to the database server and running a test program.
- **System Requirements.** Check that your system meets all requirements. For details, see "Application Server Requirements" on page 2-6.

- **Compiling to C code.** Dynamic 4GL features introduced in Version 3.0 do not support compilation to C code. If you are not using these newer features, compilation to C code is still possible.
- **Building the p-code runner.** The installation script creates a default p-code runner named **fglrun**. To create your own statically-linked runner, refer to the *Informix Dynamic 4GL User Guide*.
- **Installing both packages on the same system.** If you are installing both the development and runtime packages on the same UNIX system, create two separate directories and reset your **FGLDIR** environment variable to point to the package you intend to use. If you are installing both packages on Windows NT, install the packages on separate computers to avoid difficulties with the Windows NT registry.
- **Ownership of files during installation.** When you install Dynamic 4GL, the product files are frequently not owned by user **INFORMIX**. For example, the files are sometimes owned by the user ID and group of whomever installed the product. Dynamic 4GL does not require files to belong to user **INFORMIX**.

  UNIX

  If you want all files to belong to user **INFORMIX**, perform the installation with the user and group IDs set to user **INFORMIX** by issuing the command:

  ```
  su informix
  ```

  before you perform the installation.

  Alternatively, use the following command to recursively set the user and group on all files under **$FGLDIR**, for POSIX.2 compliant systems (including Solaris 2.6), after you have completed installation:

  ```
  su root -c "chown -R informix:informix $FGLDIR"
  ```

  ♦

Upgrading from a Previous Version

If you are upgrading from a previous version of Dynamic 4GL, unset all Dynamic 4GL environment variables, including the **FGLDIR** environment variable (or set **FGLDIR** to the new installation directory), before you install the upgrade.

If you are also upgrading the database or Informix Client Software Developer's Kit, perform the database and Informix Client Software Developer's Kit upgrades first. Then unset the following environment variables before you install the new version of Dynamic 4GL. These environment variables are reset for you during installation of Dynamic 4GL.

- **FGLDBS**. The application runner uses **FGLDBS** to point to the correct Dynamic 4GL libraries for the version of ESQL/C you are using. When you update the database or Informix Client Software Developer's Kit, the version changes.
- **FGLLIBSQL.** The application runner uses **FGLLIBSQL** to locate the ESQL/C libraries. If you change the path or add a new library during an update of the database server or Informix Client Software Developer's Kit, Dynamic 4GL needs to reset this environment variable in order for the runner to find the new library. If you are adding a library, use a colon to separate each library when the installation script prompts you.
- **FGLLIBSYS.** This environment variable points to the operating-system libraries that Dynamic 4GL uses to create the application runner.

Make sure that no one is using the earlier version of Dynamic 4GL during the installation of the new one and stop all Dynamic 4GL daemons. You might also want to back up the previous version of Dynamic 4GL before you install the new version.

Application Server Requirements

The following table describes the application server system requirements to install the Dynamic 4GL development and runtime software. For development purposes, your application server can also function as your front-end client.

System Component	UNIX Requirements	Windows NT Requirements
System Software	■ Supported operating system (see "Supported Operating Systems" on page 2-8) ■ ANSI-compatible C compiler (see "C Compiler Requirements" on page 2-9) ■ TCP/IP connection ■ Development version of TCP/IP installed as the default ■ Informix Client Software Developer's Kit 2.01 or later (required for GLS) *or* INFORMIX-ESQL/C 4.10 or higher (If you are using an earlier database, you can use an earlier version of ESQL/C, but features will be limited.)	■ Supported operating system (see "Supported Operating Systems" on page 2-8) ■ Compiler for Microsoft Visual C++ 4.0 or higher ■ TCP/IP Connection using the Microsoft TCP/IP stack ■ Development version of TCP/IP installed as the default ■ Informix Client Software Developer's Kit 2.10 or later (required for GLS) *or* INFORMIX-ESQL/C 7.201TE1 or higher ■ rlogin service if one is not already provided on your Windows NT system

(1 of 2)

System Component	UNIX Requirements	Windows NT Requirements
Hardware	■ Network card ■ IP address	■ Network card ■ IP address
Memory	■ 4MB to start each instance of the DVM ■ Sufficient memory for your application	■ 32 MB RAM if using Win95/98 for front-end ■ 50 MB RAM if using Windows NT for front-end
Disk Space	■ 40 MB of available disk space recommended in the **/tmp** directory or other specified directory (space is released after the installation is complete) ■ Additional 20 MB of available disk space for the Tcl/Tk interpreter if you plan to install the X11 Client on the application server	■ 98 MB of available disk space recommended in the **%TEMP%** directory or other specified directory (space is released after the installation is complete) ■ Additional 30 MB of available disk space if you plan to install the Windows Client on the application server

(2 of 2)

Warning: *Do not change the network card after Dynamic 4GL is installed. Changing the network card disables the Dynamic 4GL license information.*

Supported Operating Systems

You can install Dynamic 4GL on the operating systems listed in the following table. Some operating systems do not include Global Language Support (GLS), as the table indicates.

Operating System	Version	GLS Support
AIX	3.2.5	No
	4.2.1 and higher	Yes
DG/UX	R4.20 and higher (Intel)	Yes
HPUX	10.01 and higher	Yes
SCO UNIX	3.2.5.0.x (ELF)	Yes
Unixware	2.10.x	Yes
	7	Yes
Digital UNIX (OSF)	4.0	Yes
IRIX	5.3 (o32 format)	Yes
	6.2 and higher (n32 format)	Yes
Reliant UNIX (SINIX)	5.43 and higher	Yes
SUN	SPARC Solaris 2.5.1 and higher	Yes
	Intel Solaris 2.6 and higher	Yes
Windows	NT4.0 SP3 (Intel) with Microsoft Service Pack 3	Yes
Dynix/Ptx	4.4.2 and higher	Yes
Linux	kernel 2.0.36 and higher and glibc 2.0.7 and higher	Yes

Important: *For information about any changes to the supported operating systems, refer to the* ***machine.txt*** *file located in the* ***D4GL\PACKAGES****version (where version is the version number of Dynamic 4GL) directory of your Dynamic 4GL CD.*

C Compiler Requirements

UNIX

You must have an ANSI-compatible C compiler on the development computer before you install Dynamic 4GL. The installation script uses the C compiler to create a default runner that links the following libraries:

- System libraries
- INFORMIX-ESQL/C libraries (available in **Informix Client Software Developer's Kit**)
- Dynamic 4GL libraries

The C compiler is also required to create the p-code runner and enables you to call C language functions from your application.

The C compiler must accept the **-c** flag to produce object files and the **-o** flag to produce executable files. The C compiler must also support function prototypes and be able to compile ESQL/C programs.

If you are installing Dynamic 4GL on UNIX and do not want to use your native C compiler or you do not have a C compiler already installed, you can install the GNU C (GCC) compiler delivered with Dynamic 4GL.

If you are not using the default C compiler (which is normally cc), make sure that you set the **INFORMIXC** environment variable to the compiler you are using, such as gcc (the GNU C compiler), as well as the **FGLCC** and **CC** environment variables, as follows:

```
INFORMIXC=gcc
export INFORMIXC
```

Be sure to add the path to the C compiler to the **PATH** environment variable setting. ♦

WIN NT

For Windows NT, the C compiler is not required for installation, but you will need it during development if you want to build a custom runner. You must use the C compiler available in Microsoft Visual C++ 4.0 (or higher). ♦

Informix Client SDK and ESQL/C Requirements

Informix recommends that you use Informix Client Software Developer's Kit (Client SDK), Version 2.01 or higher. You can download **Informix Client Software Developer's Kit** from the following Web site:

```
http://www.intraware.com/informix/
```

Alternatively, you can use ESQL/C, Version 4.1 or higher, if you are not planning on using GLS. However, some features (for example, EXECUTE IMMEDIATE) are not available in ESQL/C, Version 4.1, but are available in later versions. The version of ESQL/C that you can use is dependent upon the database version you are using. If you plan to upgrade your database or version of ESQL/C, be sure to read your database server documentation before you proceed with the installation of Dynamic 4GL.

GLS

To use GLS features, you need **Informix Client Software Developer's Kit**, Version 2.01 or higher, installed prior to installing Dynamic 4GL. If you do not need the GLS libraries, **Informix Client Software Developer's Kit** is recommended but not required. ♦

Warning: *If you are using Windows NT, Version 4.0, you must use ESQL/C, Version 7.20.TE1 or higher, because earlier versions of ESQL/C, Version 7.20x, can cause system instability.*

UNIX

Installing Dynamic 4GL on UNIX

This section tells you how to install the Dynamic 4GL compiler and runtime packages on UNIX. You install Dynamic 4GL using a shell script.

Before you begin, be sure that:

- You have met all system requirements
- An ANSI-compatible C compiler is already installed
- Either Informix Informix Client Software Developer's Kit, Version 2.01 or higher, or INFORMIX-ESQL/C, Version 4.10 or higher, is already installed and properly configured (You must have Informix Client Software Developer's Kit, Version 2.01 or higher, installed if you intend to use GLS.)
- The Informix database server is running during installation

To install Dynamic 4GL on UNIX

1. Verify that ESQL/C is correctly installed and configured.

   To check the version and settings of ESQL/C, enter the command:

   ```
   esql-V
   ```

   If necessary, set **INFORMIXDIR**, **INFORMIXSERVER**, and **INFORMIXSQLHSTS**.

2. Log on as **root**.

3. Mount the Dynamic 4GL CD on your file system:

   ```
   $ mount your_cdrom_device_name /cdrom
   ```

   Depending on your system, the syntax of the mount command can be different. Check your UNIX manual. Also, depending on your operating system, the names of the files located on the CD might be in either lowercase or uppercase letters.

4. Go to the mount directory **/OS/UNIX**:

   ```
   cd /mnt/cdrom/D4GL/PACKAGES/version
   ```

   where *version* is the version number of Dynamic 4GL.

5. Run the installation script as follows:

```
$ /bin/sh ./ifx-f4gl package version system.sh
```

where *package* is the package you want to install, *version* is the version number of the package, and *system* is your operating system (including the operating-system version number). The following table lists the packages available.

Package	Description
dev	Installs the development system, including all the tools needed to compile and execute your 4GL programs
rte	Installs the runtime system libraries, allowing you to execute previously compiled Dynamic 4GL programs

If you do not have a **/tmp** directory or do not have enough space on your **/tmp** directory for Dynamic 4GL, you need to direct the installation script to another directory using the -t flag. For example:

```
$ /bin/sh ./ifx-f4gl package version system.sh -t /usr/tmp
```

6. Follow the prompts to continue with the installation.

If you see the phrase `NOT FOUND`, the installation script is looking for an item in a location other than expected. This phrase does not indicate an installation error.

You will be prompted to set the default value of **FGLDIR**, which is the directory for your Dynamic 4GL software. Set a directory for Dynamic 4GL that is separate from **Informix Client Software Developer's Kit** and not in **INFORMIXDIR**.

After Dynamic 4GL has installed all the application files, you are prompted to license the software.

7. Follow the prompts to license the software.

   To license Dynamic 4GL, you need the serial number and serial number key located on the serial number card in the License Agreement envelope you received with the software. (The serial number is the **License S/N numbe**r on the serial number card.)You also need access to the Internet and a Web browser.

   After you enter the serial number and key, display the Web site located at the following address:

   ```
   http://www.informix.com/keyissue
   ```

   Follow the Web site directions to generate an installation key and enter the key to complete the licensing. For example:

   ```
   Do you want to give the installation key now (y/n)?
   Enter the installation KEY (call your vendor to obtain it)
      > 9PF6DKCAUTCU
   License installation successful
   ```

   The following additional notes are for special cases:

   - If you answer no to the prompt, you have 30 days to enter the installation key using the following command to complete the license installation:

     ```
     licencef4GL -k
     ```

   - If you reinstall the software into the same **FGLDIR** directory (either physically or using a symbolic link,) you can avoid entering a new serial number key. However, you will need a new installation key.
   - If you have to change your serial number, you must first uninstall the current license. To do so, enter the following command:

     ```
     fglWrt -d
     ```

8. Follow the prompts to install the libraries.

   If you installed the compiler package, the script prompts you to indicate whether you want the p-code or C code libraries installed after the licensing is complete. The default is `yes` for p-code and `no` for C code. Indicate which libraries you want installed. (You will not need the C code libraries unless you are planning to compile or execute a C code version of a 4GL application. However, you can set both options to yes.) When you select the p-code libraries, the automatic installation builds a default p-code runner for you.

9. Execute the **envcomp** script to configure your environment:

   ```
   . ./envcomp
   ```

You are now ready to compile your 4GL source code with Dynamic 4GL.

Tip: *To install Dynamic 4GL from a network, copy the self-extracting files located in the directory* ***/OS/UNIX/****your_OS_name****/SELFEXTR/*** *to a directory on your application server. You must use binary transfer (8-bit) and not ASCII transfer (7-bit) mode.*

WIN NT

Installing Dynamic 4GL on Windows NT

This section tells you how to install the Dynamic 4GL compiler and runtime packages on a Microsoft Windows NT system.

Before you begin, be sure that:

- You have met all system requirements
- Either Informix Informix Client Software Developer's Kit, Version 2.01 or higher, or INFORMIX-ESQL/C, Version 4.10 or higher, is already installed (You must have Informix Client Software Developer's Kit, Version 2.01 or higher, installed if you intend to use GLS.)
- The Informix database server is running during installation
- You might also want to install a C compiler (Microsoft Visual C++ 4.0 or later) if you plan to build a customer runner

If you are planning to display your application on the same computer you are developing on, or if you are only installing the runtime package, install the Dynamic 4GL Windows Client first. The installation software looks for this client and creates icons that allow you to test if the package is correctly installed. For instructions, see Chapter 3, "Installing the Windows Client."

To install Dynamic 4GL on Windows NT

1. Verify that ESQL/C is correctly installed and configured.

   To check the ESQL/C settings, use the **SetNet** utility to attempt to connect to your database through I-Login.

   If ESQL/C is not properly configured, or you are unable to connect to the database, set **INFORMIXDIR**, **INFORMIXSERVER**, and **INFORMIXSQLHSTS**. For details, refer to your ESQL/C or Informix Client Software Developer's Kit documentation.

2. Insert the Dynamic 4GL CD into the CD-ROM drive.

3. From the **Start** menu, choose **Run**.

4. In the **Run** dialog box, enter:

   ```
   E:\D4GL\PACKAGES\ifx-f4gl package version system.exe
   ```

   where *E* represents your CD-ROM drive letter, *package* is the package you want to install, *version* is the version number of the package, and *system* is your operating system (including the operating-system version number). The following table lists the packages available.

Package	Description
dev	Installs the development system, including all the tools needed to compile and execute your 4GL programs
rte	Installs the runtime system libraries, allowing you to execute previously compiled Dynamic 4GL programs

   The Installation Wizard appears.

5. Click **Continue** and choose from the following installation options:

   - **Automatic search of an existing Informix version.** This option searches the database registry for the directory where your Informix products are installed (either the database server or Informix CLI).
   - **Specify Informix directory.** This option prompts you for the directory where your Informix products are installed.
   - **Informix is not installed.** This option indicates that no Informix product has been installed on your computer.

   Typically, you will use the **automatic search** option.

   When you choose an option, the installation program looks for the location of the Informix database and the version of ESQL/C. It also sets the **INFORMIXDIR**, **INFORMIXSQLHOSTS**, and **INFORMIXSERVER** environment variables.

   If any Informix product is found, the installation program displays a dialog box that says the installation program will install a runner for the Informix database (**fglrun**) and a runner for the nondatabase (**fglnodb**) application.

6. Click **Accept** to continue if the **INFORMIXDIR** environment variable is correctly set to the Informix product directory.

   Click **Refuse** to go back to the installation options and reset the directory path if the path to the Informix product directory is not correct.

   You will be prompted to select a destination directory for Dynamic 4GL. By default, the installation program installs the package in the **\USR\FGL2C** directory. If you want to change the directory where Dynamic 4GL is installed, click **Browse** and select a directory. A dialog box appears that indicates that you are ready to install.

7. Click **Next** to start the installation procedure.

   If no license is installed, the INFORMIX License Manager Program dialog box appears at the end of the installation procedure.

8. Enter the licensing information that the INFORMIX License Manager Program dialog box requests.

   All fields are required, except for the **Check** fields. The **Check** fields enable you to check that the serial number has been entered correctly.

Important: *Do not press* ENTER *or* RETURN *to go to the next field. The* ENTER *key validates the* ***OK*** *button and completes the license installation. Use* TAB *or the mouse to move between fields.*

- **Serial number**. You will find the serial number on the serial number card in the License Agreement envelope you received with this software. The serial number is referred to as the **License S/N** number on the serial number card.
- **Serial number key**. You will find the serial number key on the serial number card in the License Agreement envelope you received with the software.
- **Type**, **Users, FGLDIR**. These fields are generated for you when you press TAB.
- **Installation Number**. The installation number is generated for you when you press TAB.
- **Installation Key**. To access the installation key, go to the following Informix Web site:

  ```
  www.informix.com/keyissue
  ```

  You can click the icon that appears in the lower right corner of the License Manager dialog box to activate your Web browser. You will need your serial number, serial number key, and installation number to obtain the installation key.

9. Click **Finish**.

   If you have installed the development package, the program compiles the p-code libraries needed to compile 4GL to p-code.

Tip: *You enter license information only once. If you need to reinstall the product, you do not need to enter the license information again unless you removed the old* ***FGLDIR*** *directory structure. However a reinstall does require a new installation key.*

Testing Your Database Connection on Windows NT

After you have installed Dynamic 4GL and set the configuration files, you can test your database connection.

To test your database connection

1. Make sure the **stores7** database is created.
2. From the Start menu, click the Workshop Icon and create a directory separate from the location of your Dynamic 4GL files and not in **FGLDIR**.
3. If necessary, set **INFORMIXSQLHOSTS** so that you can access the database.
4. Create the database schema files.

   ```
   $ fglschema stores7
   ```

5. Set **FGLDBPATH** to include the path to the **stores7.sch** file.
6. Create a sample program named **testdbs.4gl** to test the database connection. For example:

   ```
   MAIN
      DATABASE stores7
      DISPLAY "Status: ", status
   END MAIN
   ```

   Where **stores7** is the name of the database to which you are attempting to connect.
7. Compile this program with the following command:

   ```
   fgl2p -o testdbs.42r testdbs.4gl
   ```

8. Run the program with the following command:

   ```
   fglrun testdbs.42r
   ```

   If the database server is started, you will see the following message:

   ```
   Status: 0
   ```

   This message means that the connection to the database **stores7** is running correctly.

Dynamic 4GL Directories

This section describes the directories created during the installation process. All the files are under the directory you specified during the installation, which is referenced by the environment variable **FGLDIR**.

Name	Description
bin	The executable files required when you use Dynamic 4GL
bmp	The pictures included in your 4GL programs running on X11 Clients (For the other client interfaces, consult the corresponding section.)
clients	The components that support deploying Dynamic 4GL applications on Windows and the Web
defaults	Program-specific configuration files
demo	The Dynamic 4GL demonstration programs
desi	The configuration manager for X11 Clients
etc	The configuration files and some client resources
etc/ger	A filter for the German alphabet character set
etc/iso	A filter for the ISO character set
include	Contains the **f2c** directory
include/f2c	The include files for C compilation
lib	The C libraries needed at link time when you create a new runner; also contains the 4GL libraries needed when you compile 4GL programs and modules of some Dynamic 4GL tools
lock	The data files created by clients running compiled applications (Removing this directory while applications are running leads to a failure of all the currently running 4GL programs.)

(1 of 2)

Name	Description
msg	The compiled error and runtime messages
release	The latest documentation about new features, corrected bugs, and known problems and their workarounds
src	Source files of tools and 4GL libraries (It also contains an uncompiled, readable form of the error messages contained in the **msg** directory.)
toolbars	The icons you can include in the toolbars of your applications

(2 of 2)

Licensing

Dynamic 4GL counts licenses by the number of instances, not the number of clients, that connect to the application server. This section tells you how to work with licenses for both UNIX and Windows NT. It includes the following sections:

- Working with licenses on UNIX
- Working with licenses on Windows NT
- Reactivating licenses
- Workarounds for licensing errors

UNIX

Working with Licenses on UNIX Systems

To work with licenses on UNIX, use the **fglWrt** command as follows:

`fglWrt -h`	Lists available options
`fglWrt -a see`	Indicates the current license state
`fglWrt -l`	Installs a license
`fglWrt -d`	Uninstalls a license

The user count is based on two indicators:

- The number of **fglrun** commands associated with a specific TTY for a character-based application (the application is running in ASCII mode).
- The number of graphical clients communicating with the runner for a GUI application (the application is running in graphical mode).

Applications in ASCII mode must be associated with a TTY number. You might experience errors if you try to launch your application in ASCII mode with a crontab.

Reactivating a License

If you modify settings on the application server, the license-management process might stop. If this occurs, you need to reactivate the license by uninstalling the old license and reinstalling either the same license or a new license. Be sure that no runners are in use while you reactivate the license.

A list of factors that will stop the license-management process are:

- Changing the network card on SCO, DEC, and MS Windows operating systems
- Changing the TCP/IP settings for the development version of Dynamic 4GL
- Changing the hostname of the computer
- Changing the operating system
- Changing **$FGLDIR**.

  For example, changing the name of the directory or copying the **$FGLDIR** subtree directory to another directory or file system.

Workarounds for Licensing Errors

The following table lists licensing errors and workarounds.

Error	Workaround
'Bad lock directory' error message at runtime	Delete the **$FGLDIR/lock** file.
'Invalid serial number answer' error message at runtime.	Check how many process are allowed to run simultaneously on your host. Increase this number.
Error 6018 with license error message on SCO.	With SCO computers, it is not possible to run programs directly on the console terminal. Use another TTY (telnet locally if needed).
Runtime activation does not work on Windows NT.	Before you re-enter a runtime license, delete the file **%FGLDIR%\etc\f4gl.sn.**
Error 6016: Problem reading runtime license.	Check the permissions for all the files located in the **$FGLDIR** directory. The user running an application should have read permissions on all the files and write permissions on **$FGLDIR/lock** directory. It is also possible that the **FGLDIR** environment variable is incorrectly set but the **$FGLDIR/bin** directory is correctly set in the **PATH** environment variable.
The application indicates there are no user licenses left, yet the user limit is not exceeded.	Enter the command `fglWrt -u`. Call it per cron on this system every minute.
The Windows NT application server has no network card.	Install the Microsoft Loopback Driver as a network adapter. Enter the command `fglwrt -l -1` (checks for MAC address only).

WIN NT

Working with Licenses in Windows NT

To work with licenses in Windows NT, use the **fglWrt** command in a DOS shell where the environment is properly set as follows:

`fglWrt -h`	Lists available options
`fglWrt -a see`	Indicates the current license state
`fglWrt -l`	Installs a license
`fglWrt -d`	Uninstalls a license

Alternatively, you can use the License Manager available on the Windows Client to perform these tasks. Be sure to use the TAB key and not the ENTER key to move between fields.

The user count is based on the number of remote graphical clients (**fglX11d** or **WTK**) that are communicating with the application runner.

Using the License Manager

After the license server is installed, you can use the License Manager to control the license server. You access the License Manager from the Start Menu. The License Manager contains the buttons and fields listed in the following table.

Control	Description
Start button	Starts the license service
Query button	Displays the status of the service
Stop button	Stops the service
Close button	Closes the license server control panel
License button	Registers the license
License field	License number installed on the license server
Users field	Number of licensed users
Type field	Type of license installed

(1 of 2)

Control	Description
Status light	Status of the server
Status light is red	The service is not started or is not installed
Status light is yellow	An action has been requested from the server
Status light is green	The service has been successfully started

(2 of 2)

Notes on Licensing with Windows NT

Keep the following points in mind when you manage licenses on Windows NT:

- If the license server is stopped, the active programs using it are not stopped, but no new application will be able to start. When you restart the license server, it checks which clients are still active.
- If the number of users reaches the maximum and an additional user requests a connection, the license server checks if that user has already been registered. If the user has already been registered, the license server changes the internal counter. If the user has not already been registered, the license server checks all clients to see if any client is free. If a client does not respond or signals that it has no more running applications, the license considers the client as stopped and frees its place for the requesting client.
- If there is a network or client problem, the client contacts the server upon recovery. If the maximum number of users has not already been reached, the client is allowed to proceed. The client makes approximately ten attempts per minute.
- The number of licensed users should be close to the number of actual users. This avoids overuse of the license server. An algorithm optimizes the search of vacant places, but if there is no vacancy, all clients are automatically contacted. This might unnecessarily overload the network and the license server. If there is a network problem, no users can access the running applications. Otherwise, they will be able to continue when the network recovers.

Chapter

3

Installing the Windows Client

Windows Client Requirements 3-4
Front-End Client Requirements 3-4
Application Server Requirements 3-5

Installing the Windows Client 3-5
Installing the Windows Client on a Network 3-6

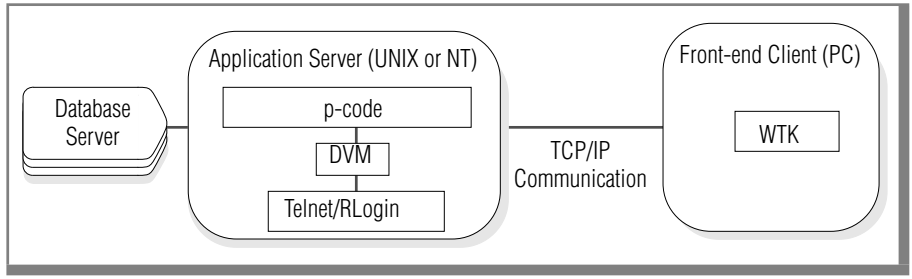
Configuring the Windows Client 3-7

In This Chapter

This chapter tells you how to install the Windows Client. The Windows Client consists of the WTK interpreter named 4GL Server.

Figure 3-1 shows the expanded architecture for the Windows Client.

Figure 3-1
Windows Client Architecture



Windows Client Requirements

This section covers the hardware and software requirements for the Windows Client. Before you begin installation, verify that your system setup meets these requirements.

Front-End Client Requirements

The front-end client is the computer that displays the application. The following requirements are for the front-end client:

- A PC running Windows 3.1, 95, 98, or NT
- TCP/IP connection
- Microsoft TCP/IP stack

  You must use the Microsoft TCP/IP stack. Do not use a third-party TCP/IP stack.

WIN 3.1

- If you are using Windows 3.1, install the WIN32S extension and Microsoft TCP/IP 32 stack.

  This software is included with the Dynamic 4GL media. To install the WIN32S extension, execute the file **setup.exe** located in the **\WINDOWS\UTIL\WIN32S\DISK1** directory on the CD. To install the Microsoft TCP/IP 32 stack, read the installation instructions in the **readme.txt** located in the **\WINDOWS\UTIL\TCPIP32\TCPIP32** directory on the CD. ♦
- Disable any memory-manager software (such as Emm386) running on the front-end client.

  Memory-manager software reduces the Windows display speed of Dynamic 4GL applications.
- For best results, the front-end client monitor should be able to display at least 256 colors.

Application Server Requirements

The application server is the computer that is running Dynamic 4GL. The Windows Client uses the WTK-Rlogin utility to connect to the application server. For the connection to occur, an rlogin service must be running on the application server. The rlogin service must also be running during installation of the Windows Client if you want to check your connections.

UNIX

If the application server is running on UNIX, an rlogin service is already installed. Be sure to activate the rlogin service. This service is used with the rlogin program to display the application in the Windows Client. ♦

WIN NT

If the application server is running on Windows NT, you need to install an rlogin service. Unlike UNIX, Windows NT does not have a built-in rlogin service. You must purchase it separately. ♦

Installing the Windows Client

The following installation process installs the Windows Client (the WTK interpreter). When the installation is complete, three program-group icons will be available from the Start menu:

- **4GL Server**
- **Add-Rlogin**
- **Configuration Manager**

Important: *You can install the client display server in the same directory as the Dynamic 4GL compiler. However, if you expect to install a compiler license and a runtime license for Dynamic 4GL on the same computer, you should keep the display server code separate. For details about licensing, see* *"Licensing" on page 2-20*.

To install the Windows Client

1. Close all applications.
2. Insert the Dynamic 4GL CD.
3. Change to the **\CLIWTK\PACKAGES***version* (where *version* is the version number of the WTK Client) directory.
4. Run **setup.exe.**

5. Follow the prompts.

   When prompted, install the 4GL Server. If you have an older version of the Windows Client installed, you are prompted to replace the older version. If you are doing an update, the same group is used and the files **rhosts, locals.tcl**, and **termuser.tcl** are not overwritten.

   When prompted, specify the directory where you want to install the application. By default, the Windows Client installs itself in the **\I4glsrv** directory on the partition where Windows is installed.

   When prompted, enter the name of the program group to store the new icons. To use a different group name when doing an update, the previous version must be uninstalled first.

Important: *Some software might be incompatible with the Windows Client, such as video drivers, networks drivers, printers, spoolers, memory-management programs, and Ethernet drivers. If an error occurs during the installation, try to install the Windows Client with a minimum of these programs started. You can disable software in your system* ***autoexec.bat*** *or* ***config.sys*** *files.*

Installing the Windows Client on a Network

The Windows Client binaries and data files can be installed from a central network server. You might want to install from a network server if you have workstations without hard disks or want to facilitate updating the client. Network file servers must be running Windows NT, Windows 95, Windows 98, or Windows 3.11.

For a client workstation installation, execute the application **newgrp.exe**, located in the **bin** subdirectory of the Windows Client.

For example, if a Windows Client is installed in the **f:\fgl2cusr** directory, users on the network that want to install a copy of the Windows Client can mount this directory as a shared drive. They can then execute **bin\newgrp.exe** to install the Windows Client program group.

When installing over a network, the uninstall icon is not included. To delete the Windows Client from a remote computer, you must delete the icons and the folder manually.

Configuring the Windows Client

After you install the Windows Client, you can configure it. You need to configure both the application server and the front-end client (where the WTK is located).

To configure the application server for the Windows Client

1. Start the rlogin service on the application server.

   If necessary, configure the rlogin service for the user and set up **SetNet 32** for this user on the application server.

2. Check that the following variables are set in **FGLDIR\etc\FGLPROFILE** on the application server:

   ```
   fglrun.database.listvar = "CC8BITLEVEL COLLCHAR CONRETRY
   CONTIME DBANSIWARN DBDATE DBLANG DBMONEY DBNLS DBPATH
   DBTEMP DBTIME DELIMIDENT ESQLMF FET_BUFF_SIZE GL_DATE
   GL_DATETIME INFORMIXDIR INFORMIXSERVER INFORMIXSQLHOSTS
   LANG LC_COLLATE LC_CTYPE LC_MONETARY LC_NUMERIC LC_TIME
   DBALSBC DBAPICODE DBASCIIBC DBCENTURY DBCODESET DBCONNECT
   DBCSCONV DBCSOVERRIDE DBCSWIDTH DBFLTMSK DBMONEYSCALE
   DBSS2 DBSS3"
   ```

3. Copy **FGLPROFILE** to ***username*.prf** (where *username* is the name of the user) and make the following changes to the *username***.prf** file:

   ```
   fglrun.remote.envvar = "REMOTEADDRESS"
   ```

   The next two lines of the following code must be uncommented:

   ```
   fglrun.setenv.0 = "INFORMIXSERVER=ol_ntserver1"
   fglrun.setenv.1 = "INFORMIXHOST=ntserver1"
   ```

   You must also add these next three lines. Here, **INFORMIXDIR** must point to the location of Client SDK or ESQL/C, and **INFORMIXSQLHOSTS** must point to the application server name. Also, if you do not set **INFORMIXSERVICE**, the default server (**turbo**) is set.

   ```
   fglrun.defaultenv.0 = "INFORMIXDIR=C:\usr\Informix"
   fglrun.defaultenv.1 = "INFORMIXSQLHOSTS=\\NTSERVER1"
   fglrun.defaultenv.2 = "INFORMIXSERVICE=servicename"
   ```

4. Copy the environment file **environment.bat** to ***username*.bat** (where *username* is the name of the user) and make any necessary edits.

5. Set the **FGLPROFILE** environment variable to the full directory path of the *username***.prf** file, for example:

   ```
   c:\FGL2C\etc\username.prf)
   ```

6. Set FGLSERVER to the instance of the Windows Client:

```
set FGLSERVER=IP_address:0
```

where *IP_address* is the IP address of the front-end client where the WTK is located. (Alternatively, you can use the computer name of the front-end client. Because the WTK is located on the front-end client and each front-end client can run only one instance, the circuit number remains constant and the instance is always 0.)

7. To enable graphical display (instead of character-based display), set FGLGUI:

```
set FGLGUI=1
```

To configure the front-end client for the Windows Client

1. Create a connection for the front-end client (where the WTK is located) using the **WTK-Rlogin emulation.**

   a. Click the **Add-Rlogin** program icon available from the **Start** menu.

   b. Enter the requested information:

   - **computer**: *application_server_name*
   - **user**: *NT_user_name*
   - **terminal**: Value is automatically set.
   - **commandline**: An optional command line to be executed after the connection is successful. This line can be left blank, but it is recommended that you use it to set the environment variables and **fglrun**, or use it to run a **.bat** file.

   c. Click **OK**.

2. When the connection is made, enter your password.

   You will be in whichever directory the rlogin is configured to start you in. You might want to configure the rlogin to start you in the **%FGLDIR%** directory.

3. Test the connection by running the test program **testdbs.42r**, which you created while following the instructions in "Testing Your Database Connection on Windows NT" on page 2-18.

   If the 4GL Server dialog box title bar displays the status as 0, the connection was successful.

Chapter

4

Installing the X11 Client

X11 Client Requirements 4-4
Front-End Client Requirements 4-4
Application Server Requirements 4-4

Installing the X11 Client 4-5

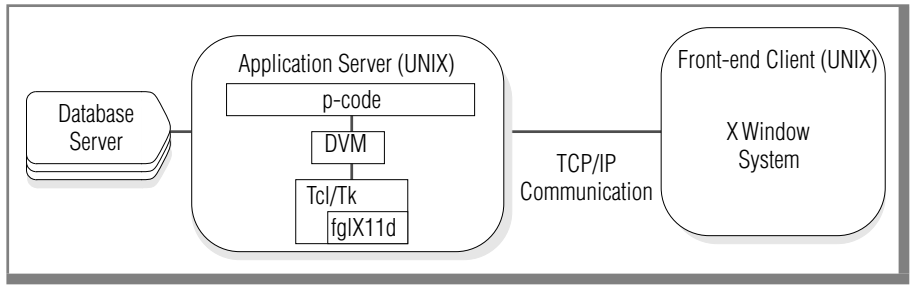
Configuring the X11 Client 4-7

In This Chapter

This chapter tells you how to install the X11 Client. The X11 Client consists of the Tcl/Tk interpreter and the X11 daemon called **fglX11d**. The X11 daemon is installed when you install either the Dynamic 4GL development package or the runtime package on UNIX. This X11 daemon is located in the **$FGLDIR/bin** directory. The Tcl/Tk interpreter is included as a separate installation on the Dynamic 4GL CD. You must use this enhanced version of the Tcl/Tk interpreter.

Figure 4-1 shows the expanded architecture for the X11 Client.

Figure 4-1
X11 Client Architecture



X11 Client Requirements

This section covers the hardware and software requirements for the X11 Client. Before you begin installation, verify that your system setup meets these requirements.

Front-End Client Requirements

The front-end client is the computer that displays the application. The following requirements are for the front-end client:

- A UNIX computer running Version X11 of the X Window System.
- The X11 Client must be installed at the same location as the DVM. If you are installing the X11 Client on a computer other than the application server, you must also install an additional unlicensed version of the compiler.
- For best results, the client computer monitor should be able to display at least 256 colors.

Application Server Requirements

The application server is the computer that is running Dynamic 4GL. Although it is possible to install the X11 Client on a separate computer, the X11 Client is typically installed on the application server. There are no additional application server requirements.

Installing the X11 Client

The installation process described below installs the Tcl/Tk interpreter. The X11 daemon is installed automatically when you install either the compiler or runtime package on UNIX. If you are installing the X11 Client on a computer other than the application server, you must first install an unlicensed version of the compiler or runtime package before you install the Tcl/Tk interpreter.

You do not need to logon as **root**. However, you need to have sufficient permissions to create the directories where you want to install the Tcl/Tk package.

Important: *You can install the client display server in the same directory as the Dynamic 4GL compiler. However, if you expect to install a compiler license and a runtime license for Dynamic 4GL on the same computer, you should keep the display server code separate*.

To install the Tcl/Tk interpreter

1. Mount the Dynamic 4GL CD on your file system:

   ```
   $ mount your_cdrom_device_name /cdrom
   ```

   Depending on your system, the syntax of the mount command can be different. Check your UNIX manual. Also, depending on your system, the names of the files located on the CD might be in either lowercase or uppercase letters.

2. Create the directory where you want to install the Tcl/Tk interpreter and change to that directory.

   For example:

   ```
   $ cd /usr/local
   $ mkdir tcl-tk
   $ cd tcl-tk
   ```

3. Unzip the tar file for the Tcl/Tk interpreter by entering the command:

   ```
   $ gzip -d/cdrom_path/CLIXTK/PACKAGES/
      ifx-tcltk-version-system.tgz | tar xvf -
   ```

   where *version* is the version number of the Tcl/Tk interpreter and *system* is the abbreviated name of your operating system followed by the version number of your operating system. You might need to review the tar files available on the CD before you enter this command.

4. Make any needed changes to the following environment variables:
   - **PATH**—add the path to the binaries
   - **TCL_LIBRARY**— specify the path to the **tcl** libraries
   - **TK_LIBRARY**—specify the path to the **tk** libraries

   The default paths are based on the location where you installed the Tcl/Tk interpreter. For example, if your installation path was **/usr/local/tcl-tk**, your defaults paths are:

```
PATH =/usr/local/tcl-tk/bin:$PATH
export PATH

TCL_LIBRARY=usr/local/tcl-tk/lib/tcl
export TCL_LIBRARY

TK_LIBRARY=usr/local/tcl-tk/lib/tk
export TK_LIBRARY
```

   Optionally, you can create a call to these settings in one of your startup files (**.profile** or **.login**).

5. Check your installation by entering the commands:

```
$ wish -v
$ wish
```

   The **-v** option displays the version of the command. After you run **wish,** a small window appears with a % prompt.

6. Enter the command:

```
% exit
```

Tip: *If you do not have a CD-ROM drive, copy the gzipped tar file to your UNIX system. You must use binary transfer (8-bit) and not ASCII transfer (7-bit) mode.*

Configuring the X11 Client

After the X11 Client is installed, you must configure the client.

To configure the X11 Client

1. Set the X11 display using either the *IP address* or the *machine_name* of the computer where the **fglX11d** daemon is located (usually your application server):

   ```
   set DISPLAY=IPaddress:0
   ```

   or

   ```
   set DISPLAY=machine_name:0
   ```

2. Set the display location using either the *IP address* or the *machine_name* of the computer where the **fglX11d** daemon is located (usually your application server):

   ```
   set FGLSERVER=machine_name:#
   export FGLSERVER
   ```

   or

   ```
   set FGLSERVER=IPaddress:#
   export FGLSERVER
   ```

   where # is the number of the **fglX11d** daemon on the server. Because each front-end client can run only one instance, the circuit number remains constant and the instance is always 0. If you are using your application server as the front-end client, # will be 0.

3. Run the X11 daemon in the background:

   ```
   fglX11d -n # &
   ```

   where # is the number of instances of the daemon.

4. Verify that the X11 daemon is running:

   ```
   ps -elf|grep fglX11d
   ```

Chapter

5

Installing the Java Client

Before Installing the Java Client 5-4
Location of Java Client Installation Components 5-4
Aliases 5-4
Pathname Variables 5-4

Java Client Requirements. 5-5
Front-End Client Requirements 5-5
Web Browser Requirements 5-6
Web Server Requirements 5-7
Application Server Requirements 5-8

Installing the Swing Components of the Java Client 5-8

Installing the Java Client on UNIX 5-9

Configuring the Java Client on UNIX 5-11

Installing the Java Client on Windows NT 5-12

Configuring the Java Client on Windows NT 5-13

Verifying CLASSPATH 5-14
Verifying the CLASSPATH Setting 5-14
Setting CLASSPATH 5-15

Configuring the Servlet Engine. 5-17
Servlet Mapping 5-17
Servlet Aliases 5-17
Apache with Apache Jserv on UNIX 5-19
Netscape with JRun on UNIX 5-19
Sun Java Web Server on Windows NT 5-21

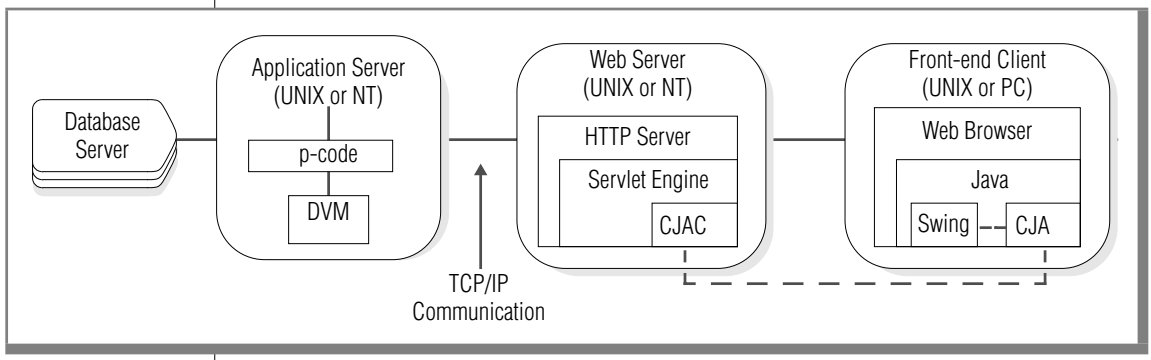
In This Chapter

This chapter tells you how to install the Java Client. The Java Client consists of:

- A Web server servlet called the Cli Java Application Connector (CJAC) that handles communication between the client Web browser and the runner.
- The Cli Java Applet (CJA) that resides on the client computer and communicates with CJAC to display information and support user input.
- A jar (Java Archive) file called **swingall.jar** for the client computer that contains the Swing library required to display the Java Client.

Figure 5-1 shows the expanded architecture for the Java Client.

Figure 5-1
Java Client Architecture



You can choose to collapse this architecture. For example, the Web server and application server can reside on the same computer.

Before Installing the Java Client

Before you install the Java Client, you should familiarize yourself with the location of the installation components, the term *alias*, and the pathname variables that are used in these installation instructions.

Location of Java Client Installation Components

The installation components for the Java Client are available in the **/CLIJAVA** directory on the CD. The directory contains the following subdirectories:

```
DOC
PACKAGE
SWING
```

Aliases

An *alias* (or virtual directory) is a name you define as a substitute for a real pathname. An alias called **clijava** can point to any directory on your Web server. The advantage of using an alias is that the full path is invisible to the user.

For more information on aliases and how to use them, refer to your Web server documentation.

Pathname Variables

The following table lists pathname variables that are used in these instructions for installing and configuring the Java Client.

Pathname Variable	Description
web_server	The Web server IP address or its DNS equivalent
web_server_port	The port on which your Web server is listening. By default, this port is 80.

(1 of 2)

Pathname Variable	Description
web_server_servlet_dir	The Web server servlets directory, seen as a UNIX or a Windows NT path
web_server_servlet_dir_alias	The alias for servlets directory *web_server_servlet_dir*
web_server_clijava_dir	The directory where you install the Java Client applet for download to the browser
web_server_clijava_dir_alias	The alias for *web_server_clijava_dir*

(2 of 2)

Tip: *Servlet and applet names are case sensitive.*

Java Client Requirements

This section covers the hardware and software requirements for the Java Client. Before you begin installation, verify that your system setup meets these requirements.

Front-End Client Requirements

The front-end client is the computer that displays the application. The following requirements are for the front-end client:

- A PC or UNIX computer with TCP/IP connection
- A Java-enabled browser (or Java Appletviewer)

  The browser (or Appletviewer) must be compliant with JDK 1.1.3 or later.

Important: *You must use the Java Swing classes provided on the Dynamic 4GL product CD.*

Web Browser Requirements

The following table summarizes the supported hardware and software for the Java Client Web browser. The browser or Java Appletviewer must support JDK Version 1.1.

Platform	Version	Hardware	Software
UNIX	Various versions	■ 32 megabytes of RAM (64 megabytes recommended)	■ Netscape Communicator 4.5 (or higher) with Java Foundation Classes (Swing) 1.1 ■ If you are using an Appletviewer, be sure the Java Runtime Environment is installed.
Windows	95/98/NT	■ 32 megabytes of RAM (64 megabytes recommended) ■ Intel or compatible Pentium class CPU at 133 MHz (200 MHz recommended) ■ 30 megabytes for the browser, 12 megabytes for the Java Plug-in, and 1 megabyte for the applet	■ Netscape Communicator 4.5 (or higher) with Java Foundation Classes (Swing) 1.1 *or* Microsoft Internet Explorer 4.x with Java Foundation Classes (Swing) 1.1 ■ Instead of the Swing Java Classes, you could install the Sun Java Plug-in. The Sun Plug-in is only available for Solaris, Windows 9x, and Windows NT 4.0. ■ If you are using an Appletviewer, the Java Runtime Environment or Java Development Kit 1.1.7 is recommended.
Mac OS	PowerPC	■ 32 megabytes of RAM (64 megabytes recommended) ■ PowerPC 603e at 200 MHz (PowerPC G3 at 233 MHz recommended) ■ 20 megabytes for the MRJ (Mac OS Runtime for Java) and 1 megabyte for the applet	■ Apple MRJ (Mac OS Runtime for Java) 2.1 ■ MacOS 8.1 or later ■ Microsoft Internet Explorer 4.0

Important: *Browser size can vary widely from one operating system to the other and depends upon the options the user selects when installing the browser.*

Web Server Requirements

The following table summarizes the supported hardware and software for the Web server. You need a Web server that supports Java servlets, such as Sun's Java Web Server, Apache with Apache JServ, IIS with JRun, or Netscape with JRun.

The Web server must be installed and correctly configured before you begin installation. The Web server must be running during installation of the Java Client so that you can test the installation.

Platform	Version	Hardware	Software
UNIX	Various versions	■ 32 megabytes of RAM (64 megabytes recommended) ■ 5 megabytes for the HTTP server, 42 megabytes for the Java Development Kit (Java Runtime Environment), and 1 megabyte for the servlet	■ Web server software that supports Java servlets and is compliant with JSDK 2.0 (such as Sun's Java WebServer, Apache with Apache JServ, IIS with JRun, or Netscape server) ■ JDK/JRE 1.1.3 or later
Windows	NT	■ 32 megabytes of RAM (64 megabytes recommended) ■ Intel or compatible Pentium class CPU at 133 MHz (200 MHz recommended) ■ 5 megabytes for the HTTP server, 5 megabytes for the Java Runtime Environment, and 1 megabyte for the servlet	■ Web server software that supports Java servlets and is compliant with JSDK 2.0 (such as Apache with Apache JServ, IIS, or Netscape with JRun) ■ JDK/JRE 1.1.3 or later

Important: *Web server (HTTP server) size can vary widely from one operating system to another other and depends upon the options the administrator selects when installing the server.*

Application Server Requirements

The application server is the computer that is running Dynamic 4GL. The Java Client requires Dynamic 4GL, Version 3.0, on the application server.

Installing the Swing Components of the Java Client

Before you begin installing the Java Client, install the swing components on the front-end client where your browser is installed. The location where you will install the swing components (and the name of the swing installer file) depends on your browser environment.

- **UNIX.** Execute the shell script found in the **/CLIJAVA/PACKAGE/***version* directory on the CD:

  ```
  ifx-clijava-all-version-allos.sh
  ```

  where *version* is the version number of the Java Client. A message appears followed by a prompt asking if you want to continue installation. Choose option 4 to install the swing components.
- **Windows 95/98/NT.** Go to `E:\clijava\swing` (where *E* is the letter of your CD-ROM drive) and run the swing executable file.
- **Macintosh.** Download the Swing installer from the Sun Java Web site. Double-click the **Swing 11-Install** icon to begin the installation. The installation wizard guides you through the installation steps. At the appropriate prompt, choose the **Runtime Only** option.

Alternatively, you can choose to install this component on the Web server computer and later transfer the file to the front-end client. To do this you must first install the Web server components, as described in "Installing the Java Client on UNIX" on page 5-9 or "Installing the Java Client on Windows NT" on page 5-12. After you install the Web server components, navigate to

```
http://web_server/web_server_clijava_dir/swing/index.html
```

and select the system on which you want to install the swing components.

Tip: *If you are using Netscape 6.0 or higher, you can skip this step. The swing components are already installed for you.*

Installing the Java Client on UNIX

This section contains instructions for installing the Java Client application server and Web server components on UNIX.

These instructions assume that you are installing the Java Client on a computer that acts as both application server and Web server. To install the Web server on a computer separate from the application server, you must perform the installation on both the application server and the separate Web server, and respond to the prompts that indicate the separate installations so that appropriate components are installed on each computer.

Tip: *If* ***jar*** *is not a command of your system, check your JDK or JRE installation and the current environment. This command is part of the standard JDK or JRE package.*

To install the Java Client on UNIX

1. Review application server, Web server, and front-end client requirements.
2. Install a Web server and configure it for servlets.

   To avoid permissions problems, stay logged on as **root** to complete the installation.
3. Install the Java Client application server components by executing the shell script found in the **/CLIJAVA/PACKAGE/***version* directory on the CD:

   ```
   ifx-clijava-all-version-allos.sh
   ```

   where *version* is the version number of the Java Client.

   If the application server and Web server are on the same computer, choose option 3; otherwise choose option 1 to install only the application server components.

   The installation shell script uses your **FGLDIR** setting as the default directory. If you choose a directory other than **FGLDIR**, you must place the components in the proper **FGLDIR** directories after installation.

   If you choose option 3 to install both the application server and Web server components on the same computer, skip the next step and proceed to "Configuring the Java Client on UNIX" on page 5-11.

4. If you are installing the Web server components on a separate computer, execute the installation shell script on that computer and follow the prompts to install the Java Client Web server components.

   You will be prompted for the directory where you want to install the first group of files. Two main groups of files are installed on the Web server:

   - Some files go into the documents directory of the Web server (often named *htdocs*). Included among these files are the CJA applet, downloaded by a browser connecting to the Web server, and the sample HTML pages that call demonstration applications.
   - The remaining files, including the CJAC servlet, go into the servlets directory.

5. Enter the root directory of your documents directory on your Web server.

   This value is used to define *web_server_clijava_dir*. If possible, use the default value as it will later make configuring the server easier.

6. Enter the location in which to install the CJAC servlet and related files.

   This value is *web_server_servlet_dir*.

   You have now completed the installation and will see the following banner:

```
############################################################
      End of the installation process
############################################################
```

Configuring the Java Client on UNIX

This section tells you how to configure and test the Java Client on UNIX.

To configure the Java Client on UNIX

1. Start the browser on the client computer and navigate to:

   ```
   http://web_server_dir:web_server_port/clijava/
      res_test_swing.html)
   ```

2. Go to *web_server_servlet_dir* on the Web server and enter:

   ```
   jar -xvf cjac.jar
   ```

   You should see the following directory structure under *web_server_servlet_dir*:

   ```
   com
   lib
   fglTestServlet.class
   cjac.jar
   META-INF
   ```

3. Configure the servlet engine on your Web server to recognize the Java Client (CJAC) servlet:

   a. Verify that your Web server supports JSDK 2.0 servlets.

   b. If your servlet engine can run target files mapped to the URL alias `/servlets/`, (plural) proceed to step **c** to create an alias.

      If the engine maps `/servlet/` (singular) or some other string to the invoker, assign a new style according to the directions in your Web server documentation. For Netscape, use the JRun administration tool to set up the mapping.

   c. Create an alias for the sample servlet, **fglTestServlet** (alias **TestServlet**). The servlet is located in your servlets directory upon unjarring **cjac.jar**.

   d. Create a second alias for **CJAC.class** (alias **cjac**). The **CJAC.class** file resides in the following directory after you unjar the **cjac.jar** file:

      servlet_directory`/com.informix.communication.CJAC.class`

4. Verify that your **CLASSPATH** environment setting in the servlet engine includes pointers to the **jsdk20.jar**, JDK classes, classes specific to your servlet engine.
5. Test the installation of the Java Client.
   a. Test the servlet capabilities of your Web server by entering the following URL in the browser on the client:
      `http://web_server:web_server_port/servlets/TestServlet`
   b. Test the CJAC servlet parameter by entering the following URL in the browser on the client:
      `http://web_server:web_server_port/servlets/cjac?TEST`

   If you get an error page, it is likely that **CLASSPATH** is not properly set. For more information, refer to your servlet engine documentation. The files not found often provide clues as to what components are missing from your **CLASSPATH** setting. For instructions on verifying **CLASSPATH**, see "Verifying CLASSPATH" on page 5-14. You might also verify that the **cjac.cnf** file is located in your ***web_server_servlets_dir*/lib** directory.

Installing the Java Client on Windows NT

This section contains instructions for installing the Java Client components on your Windows NT system.

These instructions assume that you are installing the server components of the Java Client architecture on Windows NT that acts as both application server and Web server. The client (user-interface) portion of the architecture is assumed to be installed on either UNIX, Windows 95/98/ NT, or MacIntosh.

Tip: *If **jar** is not a command of your system, check your JDK or JRE installation and the current environment. This command is part of the standard JDK or JRE package.*

To install the Java Client on Windows NT

1. From the **Start** menu, choose **Run**.
2. In the **Run** dialog box, enter:

   ```
   E:\CLIJAVA\PACKAGE\version\
      ifx-clijava-all-version-wnt0403.exe
   ```

   where *E* is your CD-ROM drive and *version* is the version number of the Java Client.

   The Installation Wizard appears.
3. Select the components you want to install and continue to follow the prompts.

   Be prepared to designate directories for the various components:

   - Application server components go into **FGLDIR**
   - Web server applet package goes in *web_server_clijava_dir*
   - Web server servlet package goes in *web_server_servlet_dir*

Configuring the Java Client on Windows NT

This section tells you how to configure and test the Java Client on Windows NT.

To configure the Java Client on Windows NT

1. Start the browser on the client computer and navigate to:

   ```
   http://web_server_dir:web_server_port/clijava/
      res_test_swing.html)
   ```
2. Go to *web_server_servlet_dir* on the Web server and enter:

   ```
   jar -xvf cjac.jar
   ```

   You should see the following directory structure under *web_server_servlet_dir*:

   ```
   com
   lib
   fglTestServlet.class
   cjac.jar
   META-INF
   ```

3. Configure the servlet engine on your Web server to recognize the Java Client (CJAC) servlet:
   a. Verify that basic (JSDK 2.0) servlets already run in your Web server.
   b. If your servlet engine can run target files mapped to the URL alias `/servlets/`, (plural) proceed to step **c** to create an alias.

      If the engine maps to `/servlet/` (singular) or some other string, assign a new style according to the directions in your Web server documentation. For Netscape, use the JRun administration tool to set up the mapping.
   c. Create an alias for the sample servlet, **fglTestServlet** (alias **TestServlet**). The servlet is located in your servlets directory upon unjarring **cjac.jar**.
   d. Create a second alias for **CJAC.class** (alias **cjac**). The **CJAC.class** file resides in the following directory after you unjar the **cjac.jar** file:

      servlet_directory`/com/informix/communication`
4. Verify that your **CLASSPATH** environment setting on the Web server includes pointers to **jsdk20.jar**, JDK classes, classes specific to your servlet engine, and **swingall.jar** (if the client is also the Web server).

Verifying CLASSPATH

After running the swing installer executable, **CLASSPATH** should be set for you. This sections tells you how to verify that **CLASSPATH** is set correctly and how to set it if it is not already set correctly.

Verifying the CLASSPATH Setting

If your Web server was capable of running servlets before you began the installation, your **CLASSPATH** should already include the correct settings. Be sure that you are using the correct version of JSDK and JDK, and at a minimum, your settings must include pointers to:

- The JSDK classes (**jsdk20.jar**), Version 2.0
- The JDK classes and source files, Version 1.1

- The classes specific to your servlet engine (for example: **ApacheJServ.jar**).
- The Swing classes, if your client computer is also your Web server (**swingall.jar**), although typically the servlet engine has its own **CLASSPATH**.

A sample setting on Windows NT follows:

```
C:\Jsdk2.0\lib\jsdk.jar;C:\jdk1.1.8\lib\classes.zip;

C:\jdk1.1.8\src;C:\JRun\lib;C:\swing-1.1\swingall.jar
```

Depending on your Web server environment, you can set **CLASSPATH** in several ways. For more information, refer to your Web server and servlet engine documentation.

Setting CLASSPATH

The following directions describe how to set the **CLASSPATH** environment variable for UNIX, Windows 9x, and Windows NT (running Microsoft Internet Explorer) client computers.

To set CLASSPATH on UNIX

1. Set the **CLASSPATH** environment variable as follows:

   ```
   CLASSPATH=$FGLDIR/clijava/lib/swingall.jar:$CLASSPATH
   export CLASSPATH
   ```

2. Place the **CLASSPATH** entry in the generic **/etc/fglprofile** file or in each user's **.fglprofile** file, so that the entry is set appropriately each time a user starts an application.
3. Restart your browser so that the setting takes effect.

   Dynamic 4GL must be running for this to work.

To set CLASSPATH on Windows 9x

1. Add a line to the **c:\autoexec.bat** file that sets **CLASSPATH** to include the **swingall.jar** file. For example:

   ```
   SET CLASSPATH=C:\swing-1.1\swingall.jar;%CLASSPATH%
   ```

2. Reboot your computer for the changes to take effect.

To set CLASSPATH on Windows NT

1. Start the **Control Panel**.
2. Double-click the **System** icon.

   The System Properties dialog box appears.
3. Click the **Environment** tab.
4. Select the **CLASSPATH** environment variable in the System Variables list box.

   If the **CLASSPATH** environment variable does not exist, click any line in the System Variables list box.

   The current user must have the appropriate rights to set environment variables. The environment variable needs to be set for all users (not just for the current user).
5. In the Variable text box, enter **CLASSPATH**.
6. Add a line to the **c:\autoexec.bat** file that sets **CLASSPATH** to include the **swingall.jar** file. For example:

   ```
   SET CLASSPATH=C:\swing-1.1\swingall.jar;%CLASSPATH%
   ```
7. In the value text box, enter **C:\swing-1.1\swingall.jar**.

   If you copied the **swingall.ja**r file in another directory, change **C:\swing-1.1** to the appropriate file location.

   If you have other **CLASSPATH** entries, separate each entry with a semicolon (;).
8. Click **Set**.
9. Click **OK**.
10. Restart Netscape Navigator or Internet Explorer, if necessary.

Configuring the Servlet Engine

If you have a properly installed Web server and servlet engine, you should already be able to run basic servlets. Your servlet engine often provides example servlets for you to verify this setup. If your Web server is not yet capable of running JSDK 2.0 servlets, refer to your Web server or servlet engine documentation for assistance.

Additional steps must be taken to enable your servlet engine to recognize and interact with components of the Java Client. The way in which these modifications are made differs depending on the Web server and servlet engine you are using. A general explanation of what must be done and examples on a few of the major platforms are in the following sections.

Servlet Mapping

When the user requests an application from the browser, the CJA applet makes a call to the Cli Java Application Connector (CJAC) servlet. CJA launches CJAC using the URL `/webserver/servlets/cjac`. Your Web server environment must be capable of recognizing such calls and invoking the servlet engine when they are received.

This allows the servlet engine to run a target file called with the name `/servlets/` as a servlet. This is often referred to as *mapping*. Most servlet engines are mapped to `/servlet/` (not `/servlets/`) by default. Therefore, it is necessary to configure the Web server to recognize calls to `/servlets/` as well. This is often done by *assigning a style*. An example using Netscape Enterprise Server appears in "Netscape with JRun on UNIX" on page 5-19.

Servlet Aliases

In addition, you must create two servlet aliases so that calls placed to the alias also call the target servlet. Your Web server or servlet engine should have a facility for adding servlet aliases.

The first alias is for the example servlet **fglTestServlet**. This servlet should be located in your servlets directory upon installation of **cjac.jar**.

fglTestServlet is not part of the functional product but is provided to allow you to verify a working environment and to illustrate the concept of servlet aliases. You need to create an alias for this servlet as follows:

Alias Name	Class Name
TestServlet	fglTestServlet

This alias enables the Web server to interpret:

```
http://myserver/servlets/TestServlet
```

as a call to invoke the servlet **fglTestServlet**.

All servlets should reside in or underneath the servlets directory. Because **fglTestServlet** is located in the servlets directory itself (for example, **Apache/servlet_directory/fglTestServlet**), you could invoke this servlet by pointing to:

```
http://myserver/servlets/fglTestServlet
```

However, it might not always be convenient or desirable to call the servlet by its filename.

The CJA calls the **CJAC.class** file using the alias **cjac**. The **CJAC.class** file resides in the **cjac.jar** file.

You must create an alias that lets the servlet engine know where to find **cjac** in the **.jar** file or on the file system under the servlets directory:

Alias Name	Class Name
cjac	com.informix.communication.CJAC

The following examples are for specific Web server environments.

Apache with Apache Jserv on UNIX

The invocation of servlets is handled by the **jserv.properties** file, which should already be configured to run servlets appropriately.

To add the needed aliases, find the **servlet.properties** file, typically located in the directory where Apache JServ is installed (for example, in **Apache/Apache Jserv**). Add the following entries:

```
servlet.cjac.code=com.fourjs.communication.CJAC
servlet.TestServlet.code=fglTestServlet
```

Netscape with JRun on UNIX

Use the JRun administration tool to configure your servlet engine for use with Java Client components. Follow the steps below.

To add the mapping to /servlets/

1. Start JRun administration.
2. Select **jse.**
3. Click the **Service Config** button.
4. Select the **Mappings** tab.
5. Click **Add.**
6. Add the following entries and click **Save**:

Virtual Path/Extension	Servlet Invoked
/servlets/	Invoker

To add the aliases

1. Start JRun administration.
2. Select **jse.**
3. Click Service Config.
4. Select the **Aliases** tab.
5. Click **Add**.
6. Add the following entry and click **Save:**

Name	Class Name
cjac	com.informix.communication.CJAC

These servlets need not be pre-loaded.

If you have not already done it, it might be necessary to *assign a style* within your Netscape Enterprise Server, as follows.

To assign a style within Netscape Enterprise Server

1. Run the Netscape Administration Server.
2. Click **Server Preferences** in the toolbar.
3. Click **View Server Settings** in the side panel.
4. Click **Configuration Styles** in the toolbar.
5. Click **Assign Style** in the side panel.
6. Fill in the requested field with `servlets/*` and choose **JRun** as the style.
7. Click **OK** and **Apply.**

   For example:

```
:8090/      servlets/*

Style:      JRun
```

Sun Java Web Server on Windows NT

Configuration for the Sun Java Web server is done using an applet in your browser. Call your Web server on port 9090, which is the Java Web Server default administration port (you can change this).

When you connect to the administration port with your browser, you will need to provide a login and password. The initial default is *admin* for both fields.

To add mapping to /servlets/

1. Select **Web Service** from the JavaWebServer Services menu.
2. Click **Manage.**
3. Click Servlet **Aliases** in the directory tree under Setup.
4. Click **Add** and add the following entries:

Alias	Servlet Invoked
/servlets/	invoker

5. Click **Save.**

To add aliases for fglTestServlet and CJAC

1. Click Servlets in the toolbar.
2. Select **Add** from the directory tree and add the following entries:

```
ServletName: TestServlet
ServletClass: fglTestServlet
```

3. Select **No** in the Bean Servlet box.
4. Click **Add.**

   You see another screen with Configuration and Properties tabs. You need not make further changes.
5. Click **Load** to complete the creation of the alias.
6. Repeat for CJAC, adding the following entries:

```
ServletName: cjac
ServletClass: com.informix.communication.CJAC
```

Chapter

6

Installing the HTML Client

Before Installing the HTML Client 6-4

HTML Client Requirements 6-4
- Front-End Client Requirements 6-4
- Web Server Requirements 6-4
- Application Server Requirements 6-5

Installing the HTML Client on UNIX. 6-5

Installing the HTML Client on Windows NT 6-7

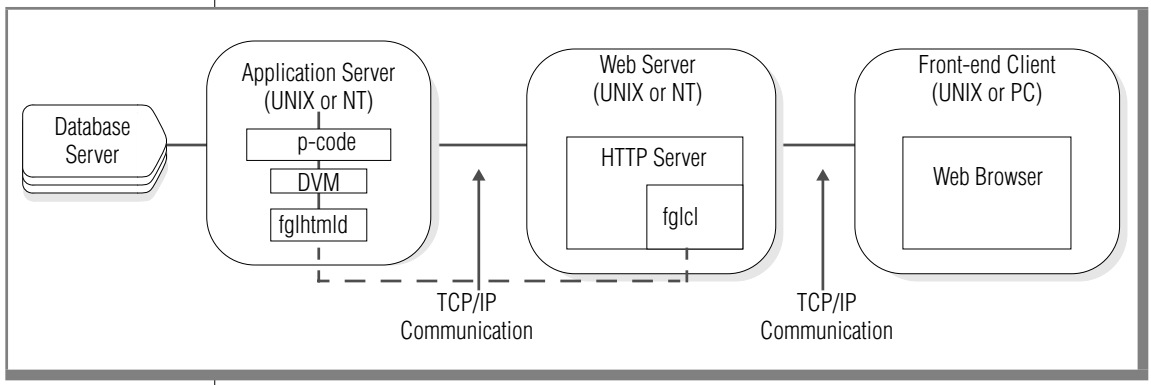
Manually Installing HTML Client Components 6-8
- Installing the HTML Client on the Web Server 6-9
- Installing the HTTP Server on the Application Server 6-10
- Installing the HTML Documentation on the Web Server. 6-11
- Installing the Example 6-12
- Configuring the HTML Client on UNIX and Windows NT 6-12

In This Chapter

This chapter tells you how to install the HTML Client. The HTML Client consists of a CGI executable called **fglcl** and an HTTP server program named **fglhtmld**.

Figure 6-1 shows the expanded architecture for the HTML Client.

Figure 6-1
HTML Client Architecture



You can choose to collapse this architecture. For example, the Web server and application server (and all four components of the HTML Client) can reside on the same computer.

Before Installing the HTML Client

Before you install the HTML Client, you should familiarize yourself with the location of the installation components on the CD.

The installation components for the HTML Client are available in the **/CLIHTML** directory on the CD. The directory contains the following subdirectories:

```
DOC
PACKAGES
PATCHES
```

HTML Client Requirements

This section covers the hardware and software requirements for the HTML Client. Before you begin the installation, verify that your system setup meets these requirements.

Front-End Client Requirements

The front-end client is the computer that displays the application. The following requirements are for the front-end client:

- A PC or UNIX computer with TCP/IP connection
- A Web browser with HTML form and table support (HTML Version 3.2 or higher)

These features are present in Microsoft Internet Explorer, Version 2.x or higher, and in Netscape Navigator, Version 2.x or higher.

Web Server Requirements

An HTTP Web server must be installed and correctly configured before you begin installation. The Web server must be configured to run CGI programs and should be running during installation of the HTML Client.

Application Server Requirements

The application server is the computer that is running Dynamic 4GL. The Dynamic 4GL runtime (or development) must be installed on the application server.

Installing the HTML Client on UNIX

This section contains instructions for installing the HTML Client components on UNIX.

To install the HTML Client on UNIX

1. Uncompress the following file to install the HTML documentation using UNIX **gzip** and **tar** utilities:

   ```
   /CLIHTML/DOC/version/ifx-clihtml-doc-version.tgz
   ```

   where *version* is the version number of the HTML Client.

2. Enter the following command to install the binaries and the example:

   ```
   sh ifx-clihtml-version-unx.sh
   ```

   where *version* is the version number of the HTML Client.

   Binaries are included for all supported UNIX systems.

3. Follow the prompts.

   You will be asked to supply the following information:

   - Whether to install the components for the application server, the Web server, or both

     If you choose both, all four components for the HTML Client will be installed on one computer. This means that both the application server and the Web server will reside on the same computer.

   - The path to your Dynamic 4GL compiler or runtime, as specified in the setting for the **FGLDIR** environment variable

     The installation adds binaries to the **bin** directory, configuration files to the **etc** directory, and message files to the **msg** directory under **FGLDIR**.

- The IP address (or hostname) of the application server

  The IP address (or hostname) is used to generate the client configuration file, **fglcl.conf**.

- The type of UNIX system on which the application server is running

  The prompt displays the system that is assumed. If you select **N**, it then displays codes for all available system types and allows you to select one.

- The root directory of the Web server. This directory is where your Web server is installed, for example, **/usr/netscape**.

- The CGI binaries directory of the Web server, for example, **/usr/netscape/cgi-bin**

- The IP address of the Web server

- The type of UNIX system on which the Web server is running

  The prompt displays the system that is assumed. If you select **N**, it then displays codes for all available system types and allows you to select one.

- Whether you want to install the documentation (HTML files that describe how to configure and use the Web deployment components)

- The location of the HTML documentation root directory on the Web server

- The default is **/var/httpd/htdocs**

  The installation will not put the documentation in this directory but will use the directory to propose a new one.

- The path to the directory in which to install the HTML documentation

  You must specify an absolute path. The default is **/var/httpd/htdocs/Cli-HTML**.

- Whether you want to install the example

- The path to the directory in which to install the example

  The default is **$FGLDIR/cli-html/example**.

- Whether you want to install the release notes
- The path to the directory in which to install the release notes

  The default is **$FGLDIR/cli-html/release**.

Tip: *To install the HTML Client from a network, copy the self-extracting files to a directory on your application server.*

Installing the HTML Client on Windows NT

This section explains how to install the HTML Client components on Windows NT.

To install the HTML Client on Windows NT

1. Execute the file named **ifx-clihtml-***version***-wnt.exe**, where version is the version number of the HTML Client.
2. Respond to each prompt and click **Next** to continue the installation.
3. Specify a location for the HTML Client on the **Choose Destination Location** page and click **Next**.

   The default is **C:\I4glsrv\Cli-HTML**. You can change the installation directory from the default, but make sure you do not specify the same directory for the compiler. (Be sure that the location specified by the **%FGLDIR%** environment variable is not the location you give for **Choose Destination Location**.)
4. Specify the type of installation you want to perform on the **Setup Type** page and click **Next**.

   You can choose from the following options:

   - **Complete installation**. Installs all the components including the HTML Client, HTTP server, documentation, and the example.
   - **Customized installation**. Displays the **Select Components** page so that you can specify which components you want to install.

5. If you chose the **Customized installation** option, specify the components you want to install on the **Select Components** page by checking the appropriate check boxes and then click **Next**.

   You can choose from the following options:

   - **HTML documentation.** These files provide configuration and usage information for deploying applications on the Web.
   - **Client and server for Windows NT.** These include HTML Client and HTTP server only.
   - **Client and server for AIX, HP-UX, IRIX, SCO, Sun Solaris (Sparc), Unixware, Linux.** This package is for UNIX systems that you can download and configure manually. Download one of these options for installation on a remote application server or Web server.
   - **Example.** For more information about the online example, see "Installing the Example" on page 6-12. The information applies to both UNIX and Windows NT.

6. Specify the program folder in which the startup icon resides and click **Next.**

   By default, the startup icon is created in the **Programs** section of the **Start** menu.

Manually Installing HTML Client Components

This section tells you how to manually perform the following tasks:

- Installing the HTML Client on the Web server
- Installing the HTTP server on the application server
- Installing the HTML documentation on the Web server
- Installing the example

To perform these tasks, you can use either the **cp** command to copy these files to their new location without removing the original files in the installation directory, or you can use the **mv** command to move these files to their new location, thereby removing the original files from the installation directory.

Installing the HTML Client on the Web Server

To install the HTML Client on the Web server, copy the following files to the directory where the Web server daemon is running. These files must reside in the **cgi-bin** directory under your main Web server. For example, on Windows NT: **c:\Inetpub\wwwroot\cgi-bin**. If this directory does not exist, create the directory and place these two files in this location:

- **fglcl** or **fglcl.exe** (the HTML Client)

  Copy this file to the **cgi-bin** directory under your main Web server directory.

- **fglcl.conf** (the configuration file for the HTML Client)

  This file contains configuration settings for each Dynamic 4GL application you are running.

  Copy this file to the **cgi-bin** directory under your main Web server directory.

The client files are initially placed in the **cgi-bin** directory of your Web server. The following code example copies each of the files from an installation directory named **/d4gl/Cli-Html** to the CGI binaries directory on a Web server named **/usr3/httpd**, and then sets appropriate file permissions (**SLS-0250** represents the directory where the HTML Client binary for the Solaris platform resides):

```
cp d4gl/Cli-Html/WebServer/cgi-bin/SLS-0250/fglcl
   usr3/httpd/cgi-bin
chmod 755 usr3/httpd/cgi-bin/fglcl

cp d4gl/Cli-Html/WebServer/cgi-bin/SLS-0250/fglcl.conf
   usr3/httpd/cgi-bin
chmod 644 usr3/httpd/cgi-bin/fglcl.conf
```

Installing the HTTP Server on the Application Server

To install the HTTP server on the application server, copy the following files from the **/AppServer** directory to the directory where your Dynamic 4GL compiler or runtime resides (as specified in the setting for the **FGLDIR** environment variable):

- **fglhtmld** (the HTTP server)

  Copy this file to the **bin** directory under **$FGLDIR**.

- **fgl2cres.web** (the resource file for the HTTP server)

  Copy this file to the **etc** directory under **$FGLDIR**.

- **fglprofile.web** (the profile for the HTTP server)

  Copy this file to the **etc** directory under **$FGLDIR**. You can also specify the **FGLPROFILE** variable to locate this file.

- **cli-html.iem** (the message file)

  Copy this file to the **msg** directory under **$FGLDIR**.

Optionally, you can also place the **fglcl** (or **fglcl.exe** for Windows NT) and **fglcl.conf** files in the **bin** directory under **$FGLDIR** as a backup for the files on the Web server.

The following code example copies each of the three files from an installation directory named **/d4gl/Cli-Html** to the directory on the application server specified by **FGLDIR**, and then sets appropriate file permissions (**SLS-0250** represents the directory where the HTML Client binary for the Solaris platform resides):

```
cp /d4gl/Cli-Html/AppServer/bin/SLS-0250/fglhtmld
   $FGLDIR/bin
chmod 755 $FGLDIR/bin/fglhtmld

cp /d4gl/Cli-Html/AppServer/etc/fgl2cres.web
   $FGLDIR/etc
chmod 644 $FGLDIR/etc/fgl2cres.web

cp /d4gl/Cli-Html/AppServer/etc/fglprofile.web
   $FGLDIR/etc
chmod 644 $FGLDIR/etc/fglprofile.web
```

Installing the HTML Documentation on the Web Server

The HTML documentation describes the **fglcl.conf** file in more detail and provides information about using the Web deployment software.

To install the HTML documentation on the Web server

1. Create a subdirectory such as **WebServer/htdocs** and extract the contents of the **ifx-clihtml-doc-***version***.tgz** file (where *version* is the version number of the HTML Client) into this directory.

   For example:

   ```
   mkdir WebServer/htdocs
   cd WebServer/htdocs
   tar -xzf ../doc.tgz
   ```

2. Create a directory named **Cli-Html** under the document root directory on your Web server and copy the documentation there. For example:

   ```
   mkdir /usr3/httpd/htdocs/Cli-Html
   cp -r . /usr3/httpd/htdocs/Cli-Html
   ```

   Be sure to name this directory **Cli-Html**. If you use another name, you need to edit the configuration files for the example program so that the example will run correctly.

3. Optionally, you can add a link from your home page to the **Cli-Html/index.html** file to make it easy to access the documentation.

Installing the Example

The **phonebook** example is a phone directory that uses the **stores7** database. The example is originally placed in the **example** directory. You can copy it to any directory.

To install the example

1. Place either the UNIX or the Windows NT version of the **phonebook** example in a directory.
2. Run **make** on UNIX or **nmake** on Windows NT and follow the on-screen instructions.
3. Enter `make install` (or `nmake install`) to install the data used in the **phonebook** example.
4. Enter `make text` to install the text version of the **phonebook** example or `make web` to install the Web version.

   The text version runs in ASCII and Windows terminals and can be deployed on the Web; however, it is not optimized for Web deployment.

   The Web version includes enhancements for Web deployment.

Configuring the HTML Client on UNIX and Windows NT

Configuring your environment to run your applications from the browser involves placing entries in the **fglcl.conf** file. This file is located in the **cgi-bin** directory of the Web server. For detailed information about configuration, see the *Informix Dynamic 4GL User Guide*.

Index

A

AIX operating system 2-8
alias 5-4
Apache with Apache Jserv 5-19
Appletviewer 5-5, 5-6
Application Server 1-6
Architecture
  Dynamic 4GL 1-5
  HTML Client 6-3
  Windows Client 3-3
  X11 Client 4-3

C

C code Intro-4, 2-4
C compiler 2-9
CC 2-9
CGI programs 6-4
CJAC. *See* Cli Java Application Connector.
cjac.cnf 5-12
cjac.jar 5-17
CJA. *See* Client Java Applet.
CLASSPATH 5-14
Cli Java Application Connector 5-3
Client Java Applet 5-3
Client SDK Intro-4, 2-10
cli-html.iem 6-10
clijava 5-4

D

Database server 1-6, 2-3
Default locale Intro-5
dev package 2-12
DG/UX operating system 2-8
Digital UNIX 2-8
Directories
  bin 2-19
  bmp 2-19
  clients 2-19
  defaults 2-19
  demo 2-19
  desi 2-19
  etc 2-19
  etc/ger 2-19
  etc/iso 2-19
  include 2-19
  include/f2c 2-19
  lib 2-19
  lock 2-19
  msg 2-20
  release Intro-8, 2-20
  src 2-20
  TEMP 2-7
  tmp 2-7
  toolbars 2-20
  $FGLDIR 2-4
  $FGLDIR/bin 4-3
  /CLIHTML 6-4
DISPLAY 4-7
DVM. *See* Dynamic Virtual Machine.
Dynamic 4GL architecture 1-5
Dynamic 4GL directory 2-3
Dynamic Virtual Machine 1-6
Dynix/Ptx operating system 2-8

E

envcomp script 2-14
Environment variables 2-15
  CC 2-9
  CLASSPATH 5-14
  DISPLAY 4-7
  FGLCC 2-9
  FGLDBS 2-5
  FGLDIR 2-3, 2-4, 2-5, 2-12, 2-19, 5-9
  FGLGUI 3-8
  FGLLIBSQL 2-5
  FGLLIBSYS 2-5
  FGLPROFILE 3-7
  FGLSERVER 3-8, 4-7
  INFORMIXC 2-9
  INFORMIXDIR 2-3, 2-11, 2-12, 2-15
  INFORMIXSERVER 2-11
  INFORMIXSQLHSTS 2-11, 2-15
  PATH 2-9, 4-6
  TCL_LIBRARY 4-6
  TK_LIBRARY 4-6
en_us.8859-1 locale Intro-5
ESQL/C Intro-4, 2-9, 2-10

F

fgl2cres.web 6-10
FGLCC 2-9
fglcl 6-3, 6-9
fglcl.conf 6-9, 6-12
fglcl.exe 6-9
FGLDBS 2-5
FGLDIR 2-3, 2-4, 2-5, 2-12, 2-19, 5-9
FGLGUI 3-8
fglhtmld 6-3, 6-10
FGLLIBSQL 2-5
FGLLIBSYS 2-5
FGLPROFILE 3-7
fglprofile.web 6-10
fglrun 2-4
FGLSERVER 3-8, 4-7
fglTestServlet 5-11, 5-17
fglWrt command 2-20, 2-23
fglX11d 4-3

G

Global Language Support Intro-5, 2-10
GLS. *See* Global Language Support.
GNU C compiler 2-9

H

HPUX operating system 2-8
HTML Client
  definition of 1-7
  requirements 6-4
HTTP server program 6-3
HTTP Web server 6-4

I

Icons
  platform Intro-7
  product Intro-7
Informix Client SDK Intro-4, 2-6, 2-10
Informix ESQL/C Intro-4
INFORMIXC 2-9
INFORMIXDIR 2-3, 2-11, 2-12, 2-15
INFORMIX-ESQL/C 2-6
INFORMIXSERVER 2-11, 2-15
INFORMIXSQLHSTS 2-11, 2-15
Installation, Windows Client 3-5
IP address 2-7
IRIX operating system 2-8
ISO 8859-1 code set Intro-5

J

Java Client, definition of 1-7
Java Development Kit 5-5
Java servlets 5-7
JDK. *See* Java Development Kit.
jsdk.jar 5-12
jserv.properties 5-19

L

license
  errors 2-22
  fglWrt command 2-20, 2-23
  how to install 2-13, 2-17
  License Manager 2-23
  reactivating 2-21
  troubleshooting 2-22
  UNIX 2-20
  user count 2-20
  Windows NT 2-23
License Manager 2-23
Linux operating system 2-8
Locale Intro-5

M

Microsoft Internet Explorer 6-4

N

NetBEUI 1-4
Netscape Enterprise Server 5-20
Netscape Navigator 6-4
Netscape with JRun 5-19
Network card 2-7

O

Operating systems
  AIX 2-8
  DG/UX 2-8
  Digital UNIX 2-8
  Dynix/Ptx 2-8
  HPUX 2-8
  IRIX 2-8
  Linux 2-8
  OSF 2-8
  Reliant UNIX 2-8
  SCO UNIX 2-8
  SINIX 2-8
  Solaris 2-8
  Unixware 2-8
  Windows NT 2-8
OSF operating system 2-8

P

PATH 2-9, 4-6
p-code runner 2-4
phonebook example 6-12
Platform icons Intro-7
Product icons Intro-7

R

Reliant UNIX 2-8
rlogin service 3-5
rte package 2-12

S

SCO UNIX operating system 2-8
servlets
  aliases 5-17
  jserv.properties 5-19
  mapping 5-17
  servlet engine 5-17
  servlet.properties 5-19
  Web server 5-3
SINIX operating system 2-8
Solaris operating system 2-8
stores7 database 6-12
Sun Java Web Server 5-21
Swing library 5-3
swingall.jar 5-3

T

Tcl/Tk interpreter 2-7, 4-3, 4-5
TCL_LIBRARY 4-6
TCP/IP 2-6
TCP/IP communication 1-4
TCP/IP connection 6-4
TK_LIBRARY 4-6

U

Unixware operating system 2-8
Upgrading 2-5
user INFORMIX 2-4

W

Web browser 1-6, 6-4
Web server
  Client location 1-6
  HTML Client 6-4
Windows Client
  definition of 1-7
  disk space for 2-7
Windows NT operating system 2-8
wish 4-6
WTK interpreter 3-3

X

X Window System 1-4, 4-4
X11 Client, definition of 1-6
X11 daemon 4-3