Contents:

5. Post-Installation Tasks: Windows

6. Pre-Installation Tasks: UNIX

7. Installation Tasks: UNIX

8. Post-Installation Tasks: UNIX

E. License File

F. Error Messages

G. Platform Notes

Index

TABLE OF CONTENTS

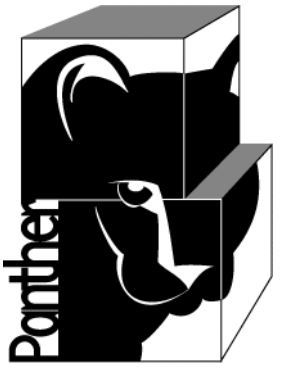
Contents:

E. License File

F. Error Messages

G. Platform Notes

Index



Panther

for IBM WebSphere
Installation Guide

Prolifics

Release 5.5

Document 0404
March 2016

Copyright

This software manual is documentation for Panther® 5.50. It is as accurate as possible at this time; however, both this manual and Panther itself are subject to revision.

Send suggestions and comments regarding this document to:

Technical Publications Manager
Prolifics, Inc.
24025 Park Sorrento, Suite 405
Calabasas, CA 91302

http://prolifics.com
support@prolifics.com
(800) 458-3313

About This Document

The *Installation Guide* contains complete instructions for installing Panther for IBM WebSphere. This includes setting up the initial configuration on server and client machines for the supported platforms. It also discusses system requirements, contents of the installation package, licensing, and linking in databases.

The guide assumes that the person doing the installation is a system administrator or someone familiar with UNIX and Windows operating systems.

This guide includes the following information:

- Pre-installation information-An overview of application architectures, the software components which make up Panther, and requirements and procedures necessary to install Panther.
- Installation instructions-Installing Panther software components on UNIX and Windows systems.
- Post-installation notes-Instructions on verifying installation and configuration, how to install the licensing software, creating new Panther executables for your installation, and information about sample applications distributed with Panther.

In addition, appendices provide information on modifications you can make to the distributed initialization files, background information on licensing including license utility programs, and the format and content of the password-enabled license file.

Documentation Website

The Panther documentation website includes manuals in HTML and PDF formats and the Java API documentation in Javadoc format. The website enables you to search the HTML files for both the manuals and the Java API.

Panther product documentation is available on the Prolifics corporate website at http://docs.prolifics.com/docs/panther/index.htm.

How to Print the Document

You can print a copy of this document from a web browser, one file at a time, by using the File→Print option on your web browser.

A PDF version of this document is available from the Panther library page of the documentation website. You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe website at https://get.adobe.com/reader/otherversions/.

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
Ctrl+Tab	Indicates that you must press two or more keys simultaneously. Initial capitalization indicates a physical key.
italics	Indicates emphasis or book titles.
UPPERCASE TEXT	Indicates Panther logical keys. *Example*: XMIT
boldface text	Indicates terms defined in the glossary.

`monospace text`	Indicates code samples, commands and their options, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <smdefs.h>`
	`chmod u+w *`
	`/usr/prolifics`
	`prolifics.ini`

`monospace italic text`	Identifies variables in code representing the information you supply. *Example*: `String` *`expr`*

`MONOSPACE UPPERCASE TEXT`	Indicates environment variables, logical operators, SQL keywords, mnemonics, or Panther constants.
	Examples:
	`CLASSPATH`
	`OR`

`{ }`	Indicates a set of choices in a syntax line. One of the items should be selected. The braces themselves should never be typed.
`|`	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

Convention	Item
[ ]	Indicates optional items in a syntax line. The brackets themselves should never be typed. *Example*: `formlib [-v] library-name [file-list]...`
. . .	Indicates one of the following in a command line: ■ That an argument can be repeated several times in a command line ■ That the statement omits additional optional arguments ■ That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. *Example*: `formlib [-v] library-name [file-list]...`
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Contact Us!

Your feedback on the Panther documentation is important to us. Send us e-mail at support@prolifics.com if you have questions or comments. In your e-mail message, please indicate that you are using the documentation for Panther 5.50.

If you have any questions about this version of Panther, or if you have problems installing and running Panther, contact Customer Support via:

- Email at support@prolifics.com
- Prolifics website at http://profapps.prolifics.com

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address and phone number

- Your company name and company address
- Your machine type
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

CHAPTER

1 General Information

In order to help you install the appropriate software components, this chapter describes the contents of the Panther installation package, application architectures, and the functionality of the Panther software components that make up your Panther distribution.

Before you begin the installation process, review the Release Notes for the latest information about Panther, including known problems.

Contents of the Installation Package

The Panther installation package consists of the items listed below. Some of these items may not be required or appropriate for your particular installation, and therefore are not included in your package. For a complete list of files included in the installation, refer to `packlist.txt` in the `notes` subdirectory of the installation.

- Panther software that includes client, application server engine, and web application broker executables, with corresponding utilities, tools, and configuration files.
- JDB, the Panther single-user database.
- Database drivers for connecting to third-party relational databases, such as DB2, ODBC, and Oracle.
- Files for C program development.

- Files for Java program development.
- Online documentation and a tutorial.
- Sample applications.

Application Architecture Description

The following list provides a description of two-tier, three-tier and web applications. Choose the appropriate architecture for your application based on your requirements.

❑	Two-tier application	The two-tier client/server model typically separates data from the logic of an application.
❑	Three-tier application	In the three-tier or enhanced client/server model, the logic of an application is distributed across servers. The backend server is known as the resource manager, and is usually a data base. The layer between client and backend server is the application server. The client is responsible for user interactions, and the application server is responsible for providing business-level services and interacting with the resource manager as needed.
❑	Web application	A web application can be a two- or three-tier application deployed on the web application server and viewed from a web browser. The web application broker works with your HTTP server software.

Installing Panther Software Components

Panther software includes the client, application server engine, and, optionally, a web application broker. The following chart describes each component, its supported platforms, and whether that component is required. Based on your chosen architecture, install the appropriate Panther software.

Panther Software Component	**Description**	**Supported Platforms**	**Required/ Optional**
Client	Provides the development and runtime client environment for building applications and components.	■ Windows ■ Solaris ■ AIX	Required
Application Server Engine	Provides access to shared application libraries and server-side components required for three-tier applications.	■ Windows ■ Solaris ■ AIX	Required
Web Application Broker	Provides the runtime environment for accessing applications through web browsers.	■ Windows ■ Solaris ■ AIX	Optional

Panther Software Component	Description	Supported Platforms	Required/ Optional
Database Drivers	Provides the interface between Panther and the supported databases. DB2 is available on Windows platforms through the ODBC driver.	*ODBC 3:* ■ Windows *Oracle:* ■ Windows ■ Solaris ■ AIX *DB2:* ■ Solaris ■ AIX	Required

The following graphics illustrate the installation process on single and multiple machine configurations.

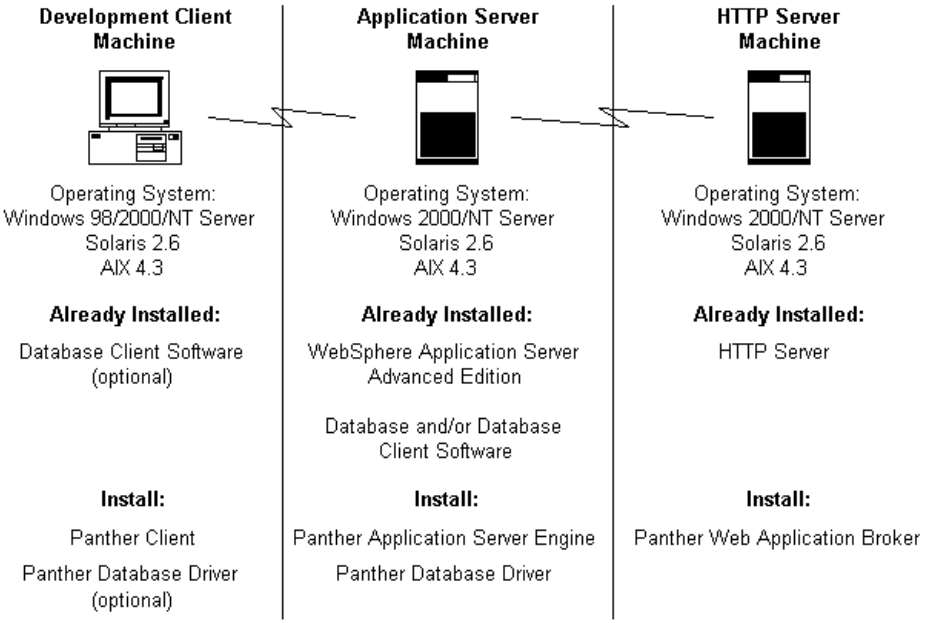


Figure 1-1 Panther Software Installed in a Networked Environment

Operating System:
Windows 2000/NT Server
Solaris 2.6
AIX 4.3

Already Installed:

WebSphere Application Server
Advanced Edition

Database and/or Database
Client Software

HTTP Server

Install:

Panther Application Server Engine

Panther Database Driver
Panther Web Application Broker

Panther Client

Figure 1-2 Panther Software Installed on a Single Machine

For More Information

After Panther is installed and configured, you can refer to the online documentation for information on specific topics.

Refer to This Guide	For Information About
Getting Started	Building an application, including setting up the client environment as well as a web application broker.
Application Development Guide	Building Panther applications.
WebSphere Developer's Studio	Building and deploying EJBs.
COM/MTS Guide	Building and deploying COM components.
Configuration Guide	Panther variables used for configuring Panther on various platforms and to your preferences and in formation on GUI resource and initialization files.
Using the Editors	The Panther graphical authoring environment.
Web Development Guide	Building and deploying a Panther web application.
Reports	Using Panther's report generation utility to build, modify and run reports.
Programming Guide	JPL commands, Panther's programming language, and using Java and C library functions in Panther.
Upgrade Guide	Upgrading from JAM or Prolifics to Panther.
`readme.*` in the `notes` subdirectory	Database-specific release notes detailing the setup of your Panther application as a database client.
`fixlist.txt` in the `notes` subdirectory	Bugs fixed in Panther.

CHAPTER

2 Installation Checklist

Use this checklist to ensure that you complete the appropriate steps to install and configure your Panther development environment for the chosen architecture.

Table 2-1 Pre-Installation Tasks

❑	Determine the application architecture.
❑	Decide which Panther components to install.
❑	Satisfy hardware and software requirements for the components you plan to install.

Table 2-2 Installation Tasks

❑	Install each Panther component.

Table 2-3 Post-Installation Tasks

❑	Submit request for permanent license (you can start using Panther with a Start-up License).
❑	Configure the client.

❑	(Three-tier only) Configure the application server engine.
❑	(Web only) Configure the web application broker.
❑	Verify installation and configuration.
❑	Look at sample applications.

When you complete the appropriate steps in the installation checklist, and set up your application environment, you are ready to build a Panther application.

CHAPTER

3 Pre-Installation Tasks: Windows

This chapter discusses the following topics:

- The hardware and software requirements for Panther software running under Windows.
- Tasks to perform before installing a web application broker.

Hardware and Software Requirements

The full installation of the Panther development environment requires the following:

- Pentium processor.
- Maximum of 65 MB of disk space for the client, 17 MB for the application server engine, and 44 MB for the web application broker. An additional 16 MB is necessary for the installation of online documentation. (However, the disk space required can be less depending on the options installed. The installer checks for adequate space.)
- Microsoft Windows server for an application server engine or web application broker; Microsoft Windows workstation or server for the client.

- 32 MB of memory. (More memory is recommended.)
- Database vendor's client and network software installed. (For more information, consult the database-specific release notes online.)
- IBM WebSphere Application Server Advanced Edition installed on the same computer as your HTTP server.
- Sun's Java Virtual Machine for EJB processing on your client PC and Sun's Java Development Kit to compile Java files. Check IBM WebSphere specifications for supported versions.
- If you are going to add your own C functions to Panther, install the Microsoft Visual C++ compiler.

We recommend that you install a Panther client, application server engine, and/or web application broker in the same directory if you install them on the same Windows machine. This allows them to share common environment settings that point to the location of required files. It also saves disk space.

To install a web application broker, an HTTP server must be configured and running on the same system. Panther supports industry favorites, and supports CGI, ISAPI, NSAPI, and Java servlet architectures. You can select one according to your preferred configuration.

Before Installing a Web Application Broker

Before installing a Panther web application broker, complete the following tasks:

- Determine the HTTP server name and the name of the CGI or ISAPI or NSAPI program directory.
- Verify that your HTTP server is running.
- Consult your system administrator or your web server administrator to determine the correct names.

- For future reference, enter the name of your HTTP server:

  _____

- Enter the name of your HTTP server program directory:

  _____

Note: This directory stores the server's gateway programs. Common names for this directory are `cgi-bin` or `scripts`.

CHAPTER

4 Installation Tasks: Windows

These instructions apply to both first-time installations and to upgrades from previous versions of JAM and Prolifics. Before beginning the installation, review the requirements and other pre-installation steps discussed in Chapter 3, “Pre-Installation Tasks: Windows.”

This chapter discusses the following topics:

- Running the setup program to install a client, an application server engine, or a web application broker.
- Installing an ODBC or Oracle database driver. DB2 support is available through ODBC.

Installing the Software

Panther is supplied in compressed form on CD-ROM along with a Windows-based setup program. You can choose to install the client, application server engine, or web application broker.

How to Run the Setup Program

1. Insert the CD-ROM in the appropriate drive.
2. If the setup program does not start automatically, choose Start→Run. In the Run dialog box, type `d:\setup` (where `d` is the letter of the drive from which you are installing).
3. Choose to install the client, application server engine, or web application broker. The setup guides you through the steps to install and configure your installation.
4. During the installation you can select a setup type.
   - Typical—Program files, samples and tutorial files, and development files.
   - Compact—Program files.
   - Custom—The software components to install, including online documentation.

   The Panther software components are as follows.
   - Program Files—Required to run Panther. Contains all configuration files, utilities, and DLLs necessary to run the Panther development environment.
   - Development Files—Optional. Only necessary if you want to add your own C code, link out certain options, or link statically with a database driver.
   - Samples and Tutorial Files—Optional. Installs the sample applications and the Tutorial.
   - Online Documentation—Optional. Because these files can be quite large, consider sharing a copy across a network if disk space is a concern. Or, use the online documentation CD instead of copying them to your PC.
5. You are prompted for a user name. Specify the user account (login) name (up to 31 characters in length). A value is required to run the development executable (`prodev`).
6. You are prompted to insert the Start-Up License diskette.
7. You are prompted for the licensing-related information. The information is used to obtain a permanent license file and includes contact information and serial number that is located on the media label.

Note: Panther software is installed with a Start-up License, provided on a separate diskette, which lets you get started. You will need to request a permanent license. Refer to page 5-2, "Obtaining Panther Licenses" when you receive your permanent license.

8. Repeat the setup program to install the other software components.
9. When the setup program is complete, reboot your system to set the new application variables.

Installing a Database Driver

After you install the Panther application server engine and/or client on a machine, you can begin installing the database driver.

Installing a database driver updates your Panther Windows client initialization file (`prol5w32.ini`) and your Panther application server file (`panther.ini`).

How to Install a Database Driver

1. Insert the CD-ROM.
2. If the setup program does not start automatically, choose Start→Run. In the Run dialog box, type `d:\setup` (where `d` is the letter of the drive from which you are installing).
3. Choose to install database drivers.
4. Select appropriate database.
5. Select from the following options:
   - Complete Install—Copies the driver files to your PC and configures the initialization file for your database version. Choose this option if you have not previously installed database driver software.

- Configuration—Updates your initialization file for another version of the database. Choose this option if you already installed the database driver and want to modify the existing database version.

6. Select or confirm the version of your database software. If your version is not listed, choose `Other`.

Because database vendors frequently update database versions, your particular database version may not be listed in the Panther install program. In this case, choose the latest version for your Panther database driver installation. Panther provides DLLs for the currently available versions of the following database client software:

Database Driver	Platform	Version
ODBC	Windows	Microsoft Open Database Connectivity version 3 Other
Oracle	Windows	Oracle Version 10i, 11i, 12c using OCI Oracle Version 10i, 11i, 12c using Pro*C Other

Note: Oracle supports two development interfaces: a C language API called OCI and an embedded SQL language, Pro*C. Most applications can use Panther's OCI or Pro*C interfaces interchangeably. Typically, Panther developers use the OCI interface unless they are linking their own custom Pro*C functions with Panther. To use Oracle's stored procedures, you must use the OCI interface.

What to Do Next

After successfully installing Panther software, there are several post-installation steps required to start using Panther. Chapter 5, “Post-Installation Tasks: Windows,” discusses the steps required to set up licensing and the environment for each of the Panther components installed, and post-installation issues and troubleshooting tips.

CHAPTER

5 Post-Installation Tasks: Windows

Post-Installation Overview

As part of post-installation, you need to perform the following general tasks, some of which are based on the architecture of your application. This chapter provides the steps required for each of these general tasks:

- Obtaining Panther Licenses on page 5-2 (A Start-up License is provided; obtain a permanent license file before the Start-up License file expires.)
- Configuring the Client on page 5-3
- Configuring the Application Server Engine on page 5-4
- Configuring the Web Application Broker on page 5-6
- Configuring the COM Development Environment on page 5-7
- Reviewing Sample Applications on page 5-9

Obtaining Panther Licenses

Your Panther software components (client, application server engine, and web application broker) require licenses. The Panther installation includes a Start-up License, provided on a diskette, that lets you use the software immediately. The temporary license, `license.dat`, is installed in the `licenses` subdirectory of the Panther installation directory. Because the license will expire, you must contact the Prolifics License Desk to receive your permanent license file. You can reach the License Desk by fax at (212) 608-6753 or by e-mail at license@prolifics.com.

During installation, the setup program prompted you to provide the information required by the License Desk for obtaining your permanent license file. It also generates license request files (`prodev.lcl` for the client, `proserv.lcl` for the application server engine, and `proweb.lcl` for the web application broker) in the `licenses` subdirectory of your Panther installation directory.

How to Obtain and Install the License File

1. Submit the license request files to the Prolifics License Desk. The License Desk returns the password-enabled file via the requested method specified during setup.
2. Complete the following step.

If you received a license file	Then
Via e-mail	Copy it to the licenses subdirectory on the machine where the software component is installed. Go to Step 4.
Via fax or mail	Copy the license request files, `.lcl`, to `.lic`. Go to Step 3.

3. Using a text editor, update `prodev.lic`, `proserv.lic`, and `proweb.lic` with the information provided in the password-enabled file that you received.

Replace text in angle brackets (and the brackets themselves) with the corresponding information provided on the `Feature` line. For example, replace <`password`> with the password provided in the license file.

4. Create the `license.dat`. Choose Start→Run, and type:

   `PantherInstallDir``\licenses\create_license`

   The `.lic` files are converted to a `license.dat` file in the `licenses` subdirectory and the Start-up License is saved as `license.bak`.

How Panther Locates the License File

Panther searches for the license file in the following order:

- The pathname given by the `LMLicenseFile` variable in your web application initialization file.
- `license.dat` in `licenses` subdirectory of the Panther installation (recommended location).
- `c:\flexlm\license.dat`.

Configuring the Client

A Panther client provides the development environment required to build two-tier, three-tier, and EJB applications. The environment must contain properly set application variables in order to run Panther. To set up the development environment, complete the following platform-specific tasks. Refer to Chapter 6, “Preparing the Development Clients,” in *Application Development Guide* for more details on setting up the client environment.

How to Configure Each Client

Set the following variables in the following initialization files: `prol5w32.ini` for Panther and `mbedit32.ini` for the menu bar editor.

- `SMFLIBS`—Specify the names of libraries to open automatically during development and at runtime.
- `CLASSPATH`—Specify the location of your Java class files.

How to Verify a Client Installation

Select Panther from the Start menu or double-click on the Panther icon in the Panther folder created during installation.

Configuring the Application Server Engine

WebSphere Application Server can be configured to support more than one server process. Each server has a directory specified as its current runtime directory. Panther provides a template initialization file, `panther.ini`, to be placed in each server directory. The template is located in the `config` directory of the Panther installation.`Panther.ini` has the following sections:

- `EJB Global`—Specify global settings for the Panther application server engine.
- `EJB Class`—Specify class settings.
- `databases`—Specify installed databases. (The installer updates these settings.)

How to Configure the Application Server

The following entries are read once, when the shared library is loaded, and are then global to all WebSphere servers.

- `LM_LICENSE_FILE`—Specify the name and location of your Panther application server engine license file or license server port.

  Note: Licensing information is required.

- `SMINITJPL`—Specify JPL file(s) to load at global initialization. The JPL file should consist only of calls to the functions `sm_slib_load` and `sm_slib_install` to make user code in DLLs available.

- `SMTPCLIENT`—Specify the type to Oracle Tuxedo client when use with Oracle Tuxedo support. Valid values are `ws` for a Oracle Tuxedo `/ws` client and `native` for a Oracle Tuxedo native client. If no value is specified, Oracle Tuxedo client support is not initialized.

- `SMTPINIT`—Specify the default arguments for the JPL command `client_init` when used with Oracle Tuxedo support.

- `SMTPJIF`—Specify the name and location of the JIF file to open on application startup when used with Oracle Tuxedo support.

- `SMVARS`—Specify the name and location of the setup file containing your Panther application variables. If the location is not specified, the default value is `smvars.bin` in *`PantherInstallDir`*`\config`.

The following variables provide settings that can be different for each WebSphere server.

- `SMBASE`—Specify the directory where Panther software is installed.

- `SMPATH`—Specify a search path for Panther application files.

- `SMINITJPL`—Specify a JPL file to load upon startup of the Panther application.

- `SMMSGS`—Specify the file that contains messages and other printable strings. If the location is not specified, the default location is *`PantherInstallDir`*`\config\msgfile.bin`.

- `SMFLIBS`—Specify the Panther libraries to open. The default value is `server.lib`.

Configuring the Web Application Broker

To ensure that your Panther application can be viewed on a web browser, complete the following platform-specific steps, most of which are applied as part of the Panther setup program. For further information on configuring web applications, refer to Chapter 2, “Web Application Setup,” in *Web Development Guide*.

Confirm that the following web application broker executables reside in your HTTP server's CGI, NSAPI, or ISAPI program directory; otherwise, copy them from the Panther `util` subdirectory:

- `websetup.exe` (Panther Web Setup Manager)
- `jwsamp.exe` (sample web application executable)

Creating a Web Application

Panther provides the Panther Web Setup Manager to guide you through configuring your customized web application. Choose Panther Web Setup Manager on the Start Menu. For more information, refer to Appendix B, “Web Setup Manager,” in *Web Development Guide*.

How to Verify your Web Application Broker Configuration and Licensing

Create a new web application to verify the web application broker installation.

1. Configure a new application by running Web Setup Manager from the Start Menu.
2. Start the web application broker from the command line of the `util` subdirectory by typing:

   ```
   monitor -install applicationName
   net start applicationName
   ```

   Refer to the `monitor` command for additional installation options.

3. Run the application in your browser:

   `http://`*`serverName/program_directory/applicationName`*

   Note: If you get the message “No Service Requested!” you were successful.

4. Stop the web application broker from the command line of the `util` subdirectory by typing:

   `net stop` *`applicationName`*

Configuring the COM Development Environment

COM components must be registered on a PC before they can be accessed by a COM application. This COM component server can be:

- A Windows workstation running COM or DCOM.
- A Windows server running COM/DCOM or MTS.

To deploy a COM component, register its DLL and type library with the system. The path for the DLL is encoded in the registration, so it is advisable to place that DLL in a specific application directory.

For Panther-built COM components, the application library containing the service components, which is created when you save the COM component in Panther, must also be available.

Template files to create COM components in Panther are:

- In the *`PantherInstallDir`*`\config` directory:
  - `PrlServer.dll`
  - `PrlServer.inf`

The required files for deployment are:

- In the *`PantherInstallDir`*`\bin` directory:

- `PrlSmCom.dll`
- `PrlTmCom.dll`
- `PrlDmCom.dll`

- Files in your application directory:
  - `server.lib`
  - *`YourApplication.dll`*

Configuring Panther with COM or DCOM

How to Deploy a COM Component

- You must register its DLL with Windows. Panther provides `Regsvr32.exe` in the `util` directory to register the COM component on a local machine.
- The application library containing the service component must be available.

The component's `.inf` file and `regcli32.bat` are provided to register the components for DCOM on a remote client machine.

For more information on deploying COM components with COM/DCOM, refer to Chapter 5, “Deploying COM Components,” in *COM/MTS Guide*.

Configuring Panther with MTS

Confirm that MTS has been installed on the Windows server.

How to Deploy COM Components with MTS

- You must register the DLL (and type library) using the Microsoft Management Console.
- The COM component must be installed into an existing component package.
- The application library containing the service component must be available.

You may also prepare components so that they can be registered remotely on client machines.

For more information on deploying COM components with MTS, refer to Chapter 5, “Deploying COM Components,” in *COM/MTS Guide*. For more information on MTS and creating or exporting component packages, see Microsoft's *MTS Documentation*.

Reviewing Sample Applications

Before you begin building your applications, you might want to review sample applications. The following sample applications are delivered with the Panther software.

- EJB Application—Sample EJB applications are provided. For more information on EJB samples, refer to Chapter B, “Sample Applications,” in *Panther for IBM WebSphere Developer's Studio*.
- Panther COM Gallery—If the COM version of Panther is installed, samples are available from the Start Menu. Instructions for installing the COM samples are included in the samples. For more information, refer to Appendix B, “COM Samples,” in *COM/MTS Guide*.
- Panther Web Gallery—Several pre-built sample applications are provided for the web.

How to Configure Web Samples

1. Confirm that the following initialization file resides in your Windows system directory; otherwise, copy it from the Panther `config` subdirectory:
   - `jwsamp.ini`
2. Check `jwsamp.ini` in the Windows directory for the correct setting for each of the following variables:
   - `AppDirectory`—Full pathname to the Panther `samples` subdirectory to provide initialization information for the sample web application

- Dispatcher—Full pathname of the dispatcher.exe program in the util subdirectory of the installation directory
- Server—Full pathname of the jserver.exe program in the util subdirectory of the installation directory
- SMBASE—Full pathname of the Panther installation directory
- PATH—Full pathname of the util subdirectory of the installation directory

3. Select either Start Panther Gallery Server or View Panther Gallery Samples from the Start menu.
4. Choose Stop Panther Gallery Server on the Start Menu when you are finished.

Troubleshooting

The information in this section provides you with guidance when you encounter error messages. The messages you may receive may be categorized as follows: graph functionality, Windows PATH settings, or online manuals.

Graph Functionality-Related Messages

Panther's graph capabilities rely upon external programs (for example, DLLs); therefore, it is possible for Panther to be working properly except for the business graph component. If you have problems with the graph functionality, check the following:

Note: The setup program normally performs all of these steps.

- Ensure that libsti.ini (distributed in the config directory) resides in the Windows directory and contains the correct paths in it for the IPT variable.
- Ensure that libsti32/64.dll (distributed in the util directory) can be found in a directory along the PATH.

Windows PATH Settings-Related Messages

If you are upgrading from JAM or Prolifics, remember that Panther's `util` directory must be on your `PATH`. Normally, this is not an issue because the installation process inserts the Panther `util` directory at the beginning of the `PATH` variable. However, the `PATH` might contain the older `util` directory. If this is the case, manually edit the `PATH` specification in the environment to remove the older `util` directory.

Online Manuals-Related Messages

Panther's online manuals are in PDF and HTML format, and require Adobe Acrobat Reader and a web browser, respectively, to view them. To verify that you can access the online documents, view the documents.

Installed documents are located in the following directory.

```
4.25:  PantherInstallationDir\docs\websphere4_25\webindex.htm
4.26+: PantherInstallationDir\docs\index.htm
```

The documents on the Windows documentation CD are located in the following directory.

```
4.25:  CD-ROMDrive:\Docs\websphere\4_25\webindex.htm
4.26+: CD-ROMDrive:\Docs\index.htm
```

Note: For more information on the online documentation, refer to Appendix A, “Panther Online Documentation.”

CHAPTER

6 Pre-Installation Tasks: UNIX

This chapter discusses the following issues to consider before installing Panther software on UNIX:

- Hardware and software requirements
- File ownership and protection
- Location of files

Hardware and Software Requirements

- The Panther client and application server engine require a maximum of 220 MB of disk space, of which 16 MB is for online documentation. The Panther web application broker requires a maximum of 50 MB of disk space.
- Running Panther under character mode has no special requirements.
- Database vendor's client and network software. For more information, consult the database-specific release notes located in the `notes` directory.
- IBM WebSphere Application Server Advanced Edition installed.
- A TCP/IP network to run the middleware or the license management software.

- A C compiler to add code in C, relink the executables, or add support for a database.
- Sun's JDK (Java Development Kit) installed to compile Java files. Check IBM WebSphere specifications for supported versions.

To run a Panther web application broker, an HTTP server must be configured and running on the same system. Panther supports some of the industry favorites, and supports CGI, ISAPI, NSAPI, and Java servlet architectures. You can select one according to your preferred configuration.

We recommend that you install the Panther web application broker and Panther software in the same directory if you installed them on the same machine. If they are not in the same directory and you need to remake the executables, set `WEBBASE` (in the `WEB PARAMETERS` section of the makefile) to the web application broker installation directory.

File Ownership and Protection

Before installing Panther software on a UNIX platform, consider the issues of file ownership and protection.

Implementing File Protection

Once they are installed, the files distributed with Panther should not be modified except under special circumstances, for example, to create a new executable. To prevent inadvertent changes to the files, we recommend that write-access to them be limited to a system administrator (or a specially created `prolifics` login), and that general Panther users be allowed only read-access.

Two suggested ways of implementing the above recommendations are:

- Login as `root` to install the files. After installation is complete, set the permissions so that only `root` can modify the files but all others can read

and/or execute them. See `chmod` in your system manual, or type `man chmod` for information on setting permissions.

- Create a dummy login ID (for example, `prolifics`), then log in as that user and perform the installation. This allows whomever has access to the `prolifics` login account to control ownership, permissions, and modifications. This approach accommodates systems for which access to the root account is tightly controlled.

Do not install Panther software into a particular user's account; it is most likely to cause maintenance problems.

Determining File Location

After deciding who is going to own the Panther files (`root` or a dummy login ID), determine where they will be installed (referred to as the Panther installation directory or `SMBASE`). Do not change this directory once it is set up because users are likely to embed the directory name in makefiles, shell scripts, and so forth. The default installation directory is `/usr/prolifics`.

If your system layout does not permit you to put the files in `/usr/prolifics`, it might allow a symbolic link from `/usr/prolifics` to the directory using the `ln -s` command.

Upgrading

If you are upgrading from a version of JAM or Prolifics, we recommend that you install Panther in a different directory from your previous installations.

Before Installing a Web Application Broker

If you plan to install a Panther web application broker on UNIX, you must create a user account called `proweb`.

How to Prepare your System for Installing a Web Application Broker

1. Add a user called `proweb` (this name assignment is required) to the `/etc/passwd` file on the machine on which your HTTP server is running. This creates a user account for the web application broker.
2. Create a web home directory for the web application broker user account. For example: `/home/proweb`.
3. Set the permissions of `proweb` so that both the Panther web application broker and the HTTP server can access the files in the web home directory in the following way:
   - Determine the name of the HTTP server user account (for example, `webmast`).
   - Create a permission `group` consisting of the web user account (`proweb`) and the HTTP server user account (`webmast`).
   - Give the group permission to access files in the web home directory (`/home/proweb`). You may need to assign the same permissions to all of the parent directories as well.
4. In your web home directory, create an `ini` subdirectory. Make sure that it has read-write permissions for everyone. Your subdirectory might look like this.

   ```
   /home/proweb/ini
   ```

CHAPTER

7 Installation Tasks: UNIX

These instructions apply to both first-time installations and to upgrades from previous versions of JAM and Prolifics. Before beginning the installation, review the requirements and other pre-installation tasks discussed in Chapter 6, “Pre-Installation Tasks: UNIX.”

This chapter discusses the following:

- Installing Panther software and the appropriate database drivers from the supplied media.
- Running the setup program to configure the installation.
- Installing the Start-up License

Note: The installation process might require several additional steps, depending on your system configuration and requirements, and whether you are installing Panther on a server or client machine.

Installing Panther

The procedure described here assumes that `/usr/prolifics` is the Panther installation directory, often referred to as `SMBASE`. If you are not installing to `/usr/prolifics`, substitute the name of the directory you are using.

Installing Panther on UNIX requires you to copy the distribution from the delivered media. The Panther installation package comes with supported database drivers for each platform.

How to Install Panther from a CD-ROM

1. Log in as `root` or with the login you devised for the installation.
2. At the command line, type the following.

   ```
   mkdir /usr/prolifics
   ```

3. Go to the `/usr/prolifics` directory by typing the following.

   ```
   cd /usr/prolifics
   ```

4. Mount the CD-ROM device as `/cdrom`.
5. In `/usr/prolifics`, to uncompress and extract the contents of the Panther distribution, type the following.

   ```
   zcat < /cdrom/CompressedTarFilename | tar -xvf -
   ```

   For example, the filenames for the Panther 4.25 distribution are as follows:

Filename	Product
`prlwascl425.tar.Z`	Panther WebSphere Edition Client
`prwassv425.tar.Z`	Panther WebSphere Edition Application Server Engine
`prlwaswb425.tar.Z`	Panther WebSphere Edition Web Application Broker

Filename	Product
`prldbs425.tar.Z`	Database driver(s)

When Panther software is loaded, your regular prompt is displayed.

Configuring the Installation

After you copy the distribution, run the setup program to configure your installation.

How to Configure the Installation

1. If you are installing the web application broker on the same machine, complete the pre-installation steps in Chapter 6.
2. Go to the `/usr/prolifics` directory; to do so, type:

   ```
   cd /usr/prolifics
   ```

3. Type `Setup`.

   The setup program guides you through the steps to configure your installation.

Once you finish running the setup program, refer to page 8-13, “Verifying the Installation.”

Using the Start-up License

Panther requires licensing in order to develop and run Panther applications. A Start-Up License is provided to allow you to begin using Panther immediately while your request for a permanent license is being processed. The Start-up License is provided on a PC diskette and on paper.

If you wish to install the license from the PC diskette, you may need to mount the diskette on a PC and then transfer the Start-up License file to the computer on which you are installing Panther software, using a transfer file utility such as ftp. Copy the `license.dat` file to `/usr/prolifics/licenses` or to the `/licenses` subdirectory of your Panther installation.

Alternatively, enter the contents of `license.dat` from the paper copy provided. For more information on installing the Panther license, refer to page 8-2, "Obtaining Panther Licensing."

Configuring a Database Driver

Database drivers need to be installed on

- Application server engine
- Development clients

Note: After you install Panther software on a machine, you can begin installing database drivers. You need to install database drivers for development clients; however, an application client at runtime does not require direct database access in a three-tier environment.

How to Configure Database Drivers

Edit the `[databases]` section of the initialization files in the `config` directory of the Panther Installation.

- Panther application server: `panther.ini`
- Panther client: `prol5unix.ini`

1. Set installed=*`Name of database(s)`*.
2. If more than one database driver is installed, set default=*`Name of default database.`*

Panther provides shared libraries for the currently available versions of the following database client software:

Database driver	Version	Installed=Name of database
DB2	DB2 Version 6	DB2_6
Oracle	Oracle Version 8.1.5 using Pro*C	oracle8iProC
	Oracle Version 8.1.5 using OCI	oracle8iOCI

Note: Oracle supports two development interfaces: a C language API called OCI and an embedded SQL language, Pro*C. Most applications can use Panther's OCI or Pro*C interfaces interchangeably. Typically, Panther developers use the OCI interface unless they are linking their own custom Pro*C functions with Panther. To use Oracle's stored procedures, you must use the OCI interface.

What to Do Next

After successfully installing Panther software, there are several post-installation steps required to start using Panther. Chapter 8 discusses the steps required to set up licensing and the environment for each of the Panther components installed, and post-installation issues and troubleshooting tips.

CHAPTER

8 Post-Installation Tasks: UNIX

After successfully installing the software, there are several post-installation steps required to start using Panther. This chapter discusses the following topics:

- Licensing
- Building Panther executables
- Setting up the environment for each of the software components installed
- Post-installation issues
- Troubleshooting tips

Post-Installation Overview

As part of post-installation, you need to perform the following general tasks for client and server installations, some of which are based on the architecture of your application. This chapter provides specific steps associated with each general task:

- Creating New Panther Executables on page 8-2
- Obtaining Panther Licensing on page 8-2 (A Start-up License is provided; obtain a permanent license file before the Start-up License file expires.)

Creating New Panther Executables

As part of the Panther installation package, a set of standard Panther executables is provided for the client, the application server engine, and the web application broker. Depending on your configuration and on the platform and database being used, you might need to create new executables.

Note: In previous versions of Panther software, you needed to create new executables to link in a different database. Panther WebSphere Edition uses shared libraries for database connectivity. The location of the shared libraries can be changed in the global initialization file.

For more information on creating new client, application server engine, and web application broker executables, refer to Appendix C, “New UNIX Executables.”

Obtaining Panther Licensing

The Panther software installation includes a Start-up License that enables you to use Panther (and run Panther applications) immediately. The temporary license, `license.dat`, is located in the `licenses` subdirectory of the Panther installation

directory. Because the license expires in 45 days, you must contact the Prolifics License Desk during this period to receive your permanent license file. Make your request as soon as possible and return to this chapter when you receive the permanent license file. You can reach the License Desk by fax at (212) 608-6753 or by e-mail at license@prolifics.com.

Obtaining a Permanent License File

You must obtain a permanent license file before the Start-up License expires to continue to develop and run Panther applications. Each development client and each server must have access to a license file.

Note: All development components—clients and servers—require licenses. In a deployed application, the application server engine and the web application broker (if applicable) require licenses, but the client does not.

During installation, the setup program prompted you to provide the License Desk with the appropriate information to obtain your permanent license file. It also generated license request files (`*.lcl`) in the `licenses` subdirectory of your Panther installation directory for each component installed on a given machine.

How to Obtain the License File

1. Submit the license request file (for each component on a given machine) to the Prolifics License Desk. The License Desk returns the password-enabled file for each component on a given machine via the requested method (as specified during setup).
2. Install the license file or files.
3. Start the license manager daemon.

Note: For more information on licensing, refer to Appendix E, “License Administration,” and to Appendix F, “License File,” for a description of the contents of a license file.

Installing the License File

Each component you install on a machine—Panther client (`prodev`), application server engine (`proserv`), and web application broker (`proweb`)—receives its own license file (`*.lic`). Install each password-enabled license file on its appropriate machine and then, via the `create_license` utility, merge them into a single `license.dat` for each machine. Follow the directions for the method of receipt.

How to Install a Permanent License File

1. Complete the following step.

If you received a license file	Then
Via e-mail	Copy it to the licenses subdirectory on the machine where the software is installed. Go to Step 3.
Via fax or mail	Copy the license request file, `.lcl`, to `.lic`. Go to Step 2.

2. Update the copy of the license file (`*.lic` file) with the information you received (via fax or mail).

   Replace text in angle brackets (and the angle brackets) with the corresponding information provided on the `Feature` line. For example, replace <`number of tokens`> and <`password`> with the appropriate information from the password-enabled license file.

   Repeat Steps 2 and 3 for each license file you received for this machine.

3. Create the `license.dat`:

   From the `licenses` directory, type:

   ```
   ./create_license
   ```

   All `.lic` files for the machine are merged into a single `license.dat` file in the `licenses` directory and the Start-up License is saved as `license.bak`.

Note: After installing the license, if your environment has set `LM_LICENSE_FILE`, you need to unset it as follows:

Bourne or Korn shell: `unset LM_LICENSE_FILE`

C Shell: `unsetenv LM_LICENSE_FILE`

Or you can explicitly set it to the license file.

How Panther Locates the License File

Panther searches for the license file in the following order:

- The pathname given by the `LMLicenseFile` variable in your web application initialization (web only).
- The pathname given by the `LM_LICENSE_FILE` line in the `.ini` file (Windows Editor only). See Windows Initialization File for more information.
- The pathname given by the `LM_LICENSE_FILE` variable.
- `license.dat` in `licenses` subdirectory of the Panther installation (recommended location).
- `license.da`.in one of the directories in `SMPATH`.

Starting the License Manager

After you create the `license.dat` file, you can start the license manager for development clients. Make sure that you are using version 5.0 or higher of the license manager daemon, `lmgrd`.

Note: The Panther application server engine and web application broker use local license files and do not require the license manager daemon. Refer to Appendix E, “License Administration.”

How to Determine the Version of `lmgrd`

- At the command line, type the following.

```
$SMBASE\util\lmver lmgrd.exe
```

How to Start the License Manager

- At the command line, type the following.

```
$SMBASE\util\lmgrd -c $SMBASE/licenses/license.dat> log &
```

where *log* is the name of a log file to which output is redirected.

Starting the license manager does not interfere with another `lmgrd` that might be running to license another application. However, if you are running `lmgrd` for previous versions of Panther, you must stop the license daemon using `lmdown` and restart `lmgrd` as described, or run `lmreread` to incorporate new Panther license daemon information.

Note: It is recommended that you add this command to the system startup scripts file so that the daemon runs automatically when the system is rebooted.

Configuring the Application Server Engine

WebSphere Application Server can be configured to support more than one server process. Each server has a directory specified as its current runtime directory. Panther provides a template initialization file, `panther.ini`, to be placed in each server directory. The template is located in the `config` directory of the Panther installation.

`Panther.ini` has the following sections:

- `EJB Global`—Specify global settings for the Panther application server engine.
- `EJB Class`—Specify class settings.
- `databases`—Specify Panther database drivers installed.

How to Configure the Application Server

The following entries are read once, when the shared library is loaded, and are then global to all WebSphere servers.

- `LM_LICENSE_FILE`—Specify the name and location of your Panther application server engine license file or license server port.

Note: Licensing information is required.

- SMINITJPL—Specify JPL file(s) to load at global initialization. The JPL file should consist only of calls to the functions sm_slib_load and sm_slib_install to make user code in shared libraries available.
- SMTPCLIENT—Specify the type to Oracle Tuxedo client when use with Oracle Tuxedo support. Valid values are ws for a Oracle Tuxedo /ws client and native for a Oracle Tuxedo native client. If no value is specified, Oracle Tuxedo client support is not initialized.
- SMTPINIT—Specify the default arguments for the JPL command client_init when used with Oracle Tuxedo support.
- SMTPJIF—Specify the name and location of the JIF file to open on application startup when used with Oracle Tuxedo support.
- SMVARS—Specify the name and location of the setup file containing your Panther application variables. If the location is not specified, the default value is smvars.bin in *PantherInstallDir*/config.

The following variables provide settings that can be different for each WebSphere server.

- SMBASE—Specify the directory where Panther software is installed.
- SMPATH—Specify a search path for Panther application files.
- SMINITJPL—Specify a JPL file to load upon startup of the Panther application.
- SMMSGS—Specify the file that contains messages and other printable strings. If the location is not specified, msgfile.bin in the installation directory will be the default.
- SMFLIBS—Specify the Panther application libraries to open. The default value is server.lib.

The following variables specify database drivers.

- default—If more than one database driver is installed, specify the name of the default database driver.
- installed—Specify the name(s) of the installed database driver(s). The following table provides the names of the drivers.

Database driver	Version	Installed=Name of database
DB2	DB2 Version 6	DB2_6
Oracle	Oracle Version 8.1.5 using Pro*C	oracle8iProC
	Oracle Version 8.1.5 using OCI	oracle8iOCI

Configuring the Web Application Broker

To ensure that your Panther application can be viewed on a web browser, complete the following platform-specific steps, most of which were applied as part of the Panther setup program. For further information on configuring web applications, refer to Chapter 2, “Web Application Setup,” in *Web Development Guide*.

How to Configure the Web Application Broker

1. Confirm that the following web application broker executables reside in your HTTP server's CGI, ISAPI, or NSAPI directory; otherwise, copy them from `$SMBASE/util`.
   - `websetup` (Panther Web Setup Manager)
   - `jwsamp` (sample application executable)
2. Confirm that the following web application broker files reside in the document root directory (consult your system administrator for the correct name) used by your HTTP server; otherwise, copy them from `$SMBASE/notes`.
   - `samples.htm` (configuration HTML files)
   - `server.htm`

- `*.gif` (graphics files)

3. Confirm that the following initialization file resides in the Panther web initialization directory (`/home/proweb/ini`); otherwise copy it from `$SMBASE/config`.
   - `jwsamp.ini`
4. Check `jwsamp.ini` in the Panther web initialization directory (`/home/proweb/ini`) for the correct setting of the following variables:
   - `AppDirectory`—Full pathname to the Panther `samples` subdirectory to provide initialization information for the sample web application
   - `Dispatcher`—Full pathname of the `dispatcher` program in the `util` subdirectory of the web installation directory
   - `Server`—Full pathname of the `jserver` program in the `util` subdirectory of the web installation directory
   - `SMBASE`—Full pathname of the Panther web installation directory
   - `PATH`—Full pathname to the `util` subdirectory of the web installation directory

Determining File Locations

Determine the following:

- Note the location of your HTTP server's program directory (such as `/usr/web/cgi-bin`) for later reference:

  The http server's program directory is:

  _____

- Check that the Panther Web Setup Manager is installed on your HTTP server and ready to run via a web browser. The default URL location is:

  `http://`*`serverName/program_directory/`*`websetup`

  Make a note of its location for later reference:

  The Setup Manager is accessed from the following URL:

  _____

Using Two Host Machines

If you are running your Panther web application broker on a machine other than the Panther application server engine, you will need to first recompile the jserver. To do this, follow these steps.

1. Go to the `/link/` subdirectory of your Panther web application broker.
2. Using an editor, such as vi or emacs, open up the file `makefile`.
3. Near the top of the file, in the section regarding which products to build, comment out all products *except* for the line indicating `JSERVER`.
4. Toward the end of the file, under Middleware Parameters, set the following parameters.

   ```
   RBLIBS=$(RBWSLIBS)
   RBWSCFLAGS=$DWSCLIENT
   #RBLIBS=$(RBNATIVELIBS)
   ```

   This builds a workstation client on the remote Panther web application broker.
5. Save and exit this file.
6. At the command line, type `make` to create the new jserver files.

Creating Your Web Application

Panther provides the Panther Web Setup Manager to guide you through configuring your customized web application. For more information, refer to Appendix B, “Web Setup Manager,” in the *Web Development Guide*.

To run the Web Setup Manager, enter the URL recorded above into a web browser.

Configuring the Client

A Panther client provides the development environment required to build two- and three-tier applications. The environment must contain properly set application variables in order to run Panther. To set up the development environment, complete the following platform-specific steps. Refer to Chapter 6, “Preparing the Development Clients,” in the *Application Development Guide* for more details on setting up the client environment.

1. For each client, copy or link the following file in the installation `config` directory to each developer's home directory:

   - `Prolifics`—X resource file used by the Panther client. Copy to each user's `app-defaults` directory (if there is one) or to the user's home directory.

2. The setup shell scripts are useful for setting Panther application variables. The setup program determined values for some of these variables, based on your system's configuration, and modified the scripts accordingly. Review `setup.sh` (Bourne or Korn shell users) and `setup.csh` (C shell users) in the application directory (if it exists) or the file `$SMBASE/config/setup.sh` (`$SMBASE/config/setup.csh`) to ensure proper settings for the following application variables:

   - `SMBASE`—Full pathname of the Panther installation directory. The installation recommends `/usr/prolifics`; therefore, `SMBASE` would be set to equal `/usr/prolifics`. If Panther was installed in a different directory, use that directory's name.

   - `PATH`—Identifies directories to search when looking for executable programs. `PATH` should include the path to Panther's `util` directory so that the system can find the development executable (`prodev`) and Panther utilities. If Panther's `util` directory is not on the `PATH`, business graphs and online help will not work.

   - `JAVA_HOME` and `CLASSPATH`—These path names are specific to Java files and event handlers and need to be set to the location where your Java

classes sit. Additionally, you can change the defaults for SMJAVAEDITOR, SMJAVAFACTORY and SMJAVALIBRARY.

- SMTERM—Specifies the type/model console the client is using. For Motif, set SMTERM to X.

  If you are running a character mode terminal with Panther, then you need to have appropriate video and key files (which instruct Panther on how to drive your terminal). Panther provides files to support several popular terminal types (which are in turn often emulated by other brands of terminals). Refer to Chapter 7, "Video File," in *Configuration Guide* for details on video files. Then examine the file $SMBASE/config/smvars. There you should find a setting of SMTERM which meets your needs.

- LM_LICENSE_FILE—Full pathname of the license file (default location is $SMBASE/licenses/license.dat).

- LD_LIBRARY_PATH—Identifies directories to search when looking for shared libraries. On AIX, use LIBPATH. LD_LIBRARY_PATH should include the path to the Motif shared library directory.

- XNLSPATH—Some implementations of Motif require you to set the XNLSPATH application variable before Panther can be successfully used. Check the documentation for your implementation of Motif to see if this is the case for you.

3. You can set SMFLIBS in setup.sh (setup.csh) to point to shared Panther application libraries in the application directory. The default setting opens only local libraries (that is, libraries in the current directory).

4. To automatically set up the environment, execute the setup shell script directly from each user's logon script.

   - Korn or Bourne shell—Append the following line to the .profile:

     applicationDir/setup.sh

     Note: If application variables are set in the user's environment for previous versions of JAM or Panther, they should be removed so that they do not interfere with the setup for this version.

5. If a user's terminal type is different from the value of SMTERM in the setup script, override that value by adding a line to the .profile or .cshrc to set SMTERM correctly. The line should immediately follow the script invocation. For example:

```
. /usr/prolifics/config/setup.sh
SMTERM=vt100
```

Alternatively, you can merge the appropriate setup script file into `.profile` or `.cshrc`, and modify the value of `SMTERM` as needed.

Overriding other application variables for a given user can be implemented in the same way.

Verifying the Installation

After you install and configure Panther software, you should verify that Panther starts up properly on client and application server engine machines. If Panther is not installed and configured correctly, an appropriate error message is displayed. Depending on the problem, the program might also terminate. Refer to Appendix G, "Error Messages," for a list of the more common error messages related to system configuration, and how to resolve them.

Client Installation

On a client machine, a correct installation is verified if you can invoke the Panther editor.

How to Verify a UNIX Client Installation

Type `prodev` at the command line.

Application Server Engine Installation

On the WebSphere Application Server machine, the Panther application server engine is loaded the first time you access a Panther-built EJB. At this time, the EJB Global section of the `panther.ini` file is initialized.

Web Application Broker Installation

On the Panther web application broker machine, verify the installation by running the Panther Gallery, and verify the configuration by running your own application.

How to Verify the Panther Web Application Broker Installation

Run the Panther Web Gallery:

1. Start the Gallery application by typing the following on the command line in the Panther `util` directory:

   ```
   monitor -start jwsamp
   ```

2. Run the Gallery by entering the following URL in your browser:

   `http://`*`serverName/program_directory`*`/jwsamp/main`

3. Stop the Gallery application by typing the following on the command line in the Panther `util` directory:

   ```
   monitor -stop jwsamp
   ```

How to Verify your Web Application Broker Configuration and Licensing

You can use the following method to verify your web application broker installation:

1. Configure a new application by running the Web Setup Manager.

2. Start the application by typing the following on the command line in the Panther `util` directory:

   `monitor -start` *`applicationName`*

3. Run the application in your browser:

   `http://`*`serverName/program_directory/applicationName`*

   Note: If you get the message “No Service Requested!” you were successful.

4. Stop the application by typing the following on the command line in the Panther `util` directory:

   `monitor -stop` *`applicationName`*

Reviewing Sample Applications

Before you begin building your applications, you might want to review sample applications. The following sample applications are delivered with Panther:

- EJB Application—Sample EJB applications are provided. For more information on EJB samples, refer to Appendix B, “Sample Applications,” in *Panther for IBM WebSphere Developer's Studio*.
- Panther Web Gallery—Provides several pre-built two- and three-tier sample applications for the web.

    a. Start the Gallery application by typing the following on the command line in the Panther `util` directory:

    ```
    monitor -start jwsamp
    ```

    b. Run the Gallery by entering the following URL in your browser:

    `http://`*`serverName/program_directory`*`/jwsamp/main`

    For example:

    ```
    http://myServer/cgi-bin/jwsamp/main
    ```

    c. Stop the Gallery application by typing the following on the command line in the Panther `util` directory:

    ```
    monitor -stop jwsamp
    ```

Post-Installation Notes

The following additional information might be required to configure and run Panther on your system.

Implementing Character-Mode Utilities

Character-mode specific utilities, `mbedit.cm` (menu bar editor) and `showkey.cm` (key translation utility), are provided in the distributed `util` directory. You may need to use these if you are running Panther using character-mode *and* your system does not have Motif installed. To ensure that directions and menu options that reference these utilities can be used and accessed, rename the executables:

1. In the `util` directory, save the Motif versions of the utilities by giving each a new extension to distinguish it from their corresponding character-mode utilities. For example:

   ```
   cp mbedit mbedit.xm
   cp showkey showkey.xm
   ```

2. Copy the character-mode utilities to the utility name having no extension. For example:

   ```
   cp mbedit.cm mbedit
   cp showkey.cm showkey
   ```

Troubleshooting

The information in this section provides you with guidance should you encounter error messages. The messages you receive may be categorized as follows: license manager, motif version number, graph functionality, or online manuals.

License Manager-Related Messages

If the license manager is not able to process the license file, an appropriate error message is displayed. Appendix G, “Error Messages,” lists error messages generated by FLEX*lm* and hints for resolving them.

If the Panther license server is not running, its clients will not be able to run.

Motif Version Number-Related Messages

From your Motif installation directory, run the following command:

```
strings libXm.a | grep -i version
```

Graph Functionality-Related Messages

Panther's graph capabilities rely upon external programs, therefore, it is possible for Panther to be working properly except for the business graph component. If you have problems with the graph functionality, check the following:

- Ensure that the `grafcap` file (distributed in the `config` directory) is located in one of the directories pointed to by `SMPATH`.
- Ensure that the files `gdsp` and `swsdrvr` (distributed in the `util` directory) can be found along the search `PATH` for executables (Panther must be able to execute these programs).

Online Manuals-Related Messages

Panther's online manuals are delivered in PDF and HTML formats. To verify that you can access the online documents, view the documents with Adobe Acrobat Reader or a web browser, respectively.

Installed documents are located in the following directory.

```
4.25:  PantherInstallationDir/docs/websphere4_25/webindex.htm
4.26+: PantherInstallationDir/docs/index.htm
```

The documents on the Windows documentation CD are located in the following directory.

```
4.25:  CD-ROMDrive:\Docs\websphere\4_25\webindex.htm
4.26+: CD-ROMDrive:\Docs\index.htm
```

Note: For more information on the online documentation, refer to Appendix A, “Panther Online Documentation.”

APPENDIX

A Panther Online Documentation

Accessing Panther Documentation

You can access the Panther documentation the following ways:

- On a local machine
- On a local intranet or network
- On the Internet
- Via the Windows online documentation CD

How to Install Documentation on a Local Machine

Panther documentation is part of the product distribution CDs for both UNIX and Windows platforms.

1. On Windows, choose the Install Documentation option when you run the installation program. The Start Menu documentation option will launch the documentation home page.
2. On UNIX, the documentation is installed as part of the product in the `docs` directory.

How to Install Documentation on an Intranet

1. Copy the documentation to your HTTP server's documentation directory or to another location on your network.
2. The URL for the home page is:

   `http://localhost/docs/index.htm`

How to Access Documentation on the Internet

1. Enter Prolifics's URL:

   `http://docs.prolifics.com/docs/panther/index.htm`

cHow to Access Documentation on the CD

In addition to the product CD, there is a documentation CD that can be accessed on a Windows machine.

1. Place the Windows online documentation CD in your CD drive.
2. The documentation home page automatically displays.

Viewing the Panther Documentation Website

When you display the documentation home page, you can access the following categories:

- What's New
- Library (for PDF and HTML versions)

- Site Maps (one for each Panther product)
- Search
- Topic Index
- Contact Information

Searching for Information

There are different methods to use to search for information.

How to Search the HTML Documentation

1. Click on the Search page, and follow the instructions for entering your search keywords.
2. Each HTML book has a Table of Contents and an Index.

How to Search the PDF Documentation

1. Within a PDF document, use Adobe's search facilities.
2. Each PDF book has a Table of Contents and an Index. If a book appears in the new PDF format, there are hyperlinks for chapters and sections.

APPENDIX

C New UNIX Executables

As part of the Panther installation package, a set of standard Panther executables is provided. Depending on the version of software, your configuration, and the platform and database being used, you might need to create new executables. Specifically, a new executable is required if you are adding another database driver or using IXI Motif libraries.

Note: Panther WebSphere Edition uses shared libraries for database connectivity, and does not require new executables to link in additional database drivers.

Create New Panther Executables

To create a new Panther executable:

1. Complete the Panther installation process, including licensing.
2. Ensure that the appropriate application variables (`SMBASE`, etc.) have been applied to your environment.
3. Go to your application directory (or create one) and copy all files from the Panther `link` subdirectory to it.
4. Edit the `makefile` in your application directory, commenting or uncommenting lines as needed to build the appropriate executables. For further information, refer to page C-2, “Specifying the Executables.”

5. Uncomment the appropriate database in the `makefile` and edit the database-specific `makevars.dbs` (where *dbs* is the extension of the database) file to choose the correct version of your database software. For further information, refer to page C-3, "Linking in the Database."

6. Type `make` at the command line to build the executable. By default, the `makefile` in the current directory is used for the `make` command.

7. The `make` process creates new Panther executables, `prodev`, `jserver`, etc. give the executable file a unique name to distinguish it from the distributed executable or others that you have built differently.

8. If you are running IXI Motif, and you rebuilt the `prorun` executable, copy the new `prorun` to `jiutil` in the Panther `util` subdirectory.

9. If you built a new web application broker executable (`jserver`), modify the `SERVER` variable in your web application's initialization file to reference the new `jserver` executable in its own directory. Do not overwrite the `jserver` executable in the `util` directory.

Specifying the Executables

To indicate the executables to build, edit the makefile in your application directory:

1. Comment or uncomment the appropriate client executables as needed (these are uncommented by default):

`PRORUN = prorun`	Runtime executable
`PRODEV = prodev`	Development executable
`RWRUN = rwrun`	Report batch utility

For platforms that have been installed with server software, comment or uncomment the appropriate server executables as needed:

`JSERVER = jserver`	Web application broker executable (uncommented by default)

`PROSERV = proserv`	Server executable for JetNet/Oracle Tuxedo (uncommented by default)
`PRODSERV = prodserv`	Debuggable server executable for JetNet/Oracle Tuxedo (commented out by default)
`PROGSERV = progserv`	Conversion server executable for JetNet/Oracle Tuxedo (commented out by default)

2. To override the value of certain application variables such as `SMBASE`, uncomment the appropriate lines in the `PARAMETERS` section.
3. If you have the Panther web application broker on the same machine as other Panther software, it is recommended that they be installed in the same directory. However, if they are not in the same directory, set `WEBBASE` (in `WEB PARAMETERS` section) to the web application broker installation directory.
4. The Panther debugger allows you to trace JPL and Panther screen events and is installed by default for clients. If you do not wish to use the debugger, comment the lines in the `DEBUGGER PARAMETERS` section.
5. You have the option of building a standalone executable and of also building a workstation client. To build either, comment or uncomment the appropriate lines in the `MIDDLEWARE PARAMETERS` section.

Linking in the Database

If you are using a database other than JDB, you must edit the `makefile` in your application directory to link in the appropriate database.

To include the appropriate database in your executables:

1. Uncomment the appropriate include statement in the `SELECT DATABASE SOFTWARE` section of the `makefile`.

   By default, JDB is uncommented; if you choose another database, you can leave JDB uncommented (and include it in your executable) or you can comment out JDB and include only your database.

2. Edit the corresponding `makevars.`*`dbs`* (where *`dbs`* is the extension of the database) file to choose the correct version of your database software.

   In the `makevars.`*`dbs`* file, verify or update the following:

   - Set the flag *`dbs`*`_INIT` to one of the following: `d`, `l`, `u`, `p`. This flag controls the handling for case sensitivity. The default is `d`. To find out what the default is for your database engine, refer to the online database-specific driver notes.
   - In the *`databaseName`* `PARAMETERS` section of `makevars`, verify your database engine's version. Uncomment the appropriate block of parameters based upon this version. Also, verify and correct the pathnames if necessary.
   - Set the flag *`dbs`*`_ENGNAME` to specify the default engine name.

   Before changing these values, refer to “Database Drivers” for additional information on database engines and on case handling.

APPENDIX

D Initialization and Databases

The installation procedure for Windows automatically modifies `PROL5W32.INI` to work with your database driver or drivers. The settings corresponding to your database and version are added to the Panther initialization file.

Note: Not all database drivers are available for all products.

For Windows Clients and Servers:

Database and Version	Settings

Microsoft Open Database Connectivity (ODBC) Version 2	`[databases]`
	`installed=odbc`
	`[dbms odbc]`
	`driver=odb2dm32.dll`
	`model=tmodb132.dll`

Microsoft Open Database Connectivity (ODBC) Version 3	`[databases]`
	`installed=odbc`
	`[dbms odbc]`
	`driver=odb3dm32.dll`
	`model=tmodb132.dll`

Database and Version	Settings

Microsoft SQL Server Version 6	`[databases]`
	`installed=sqlsrvr`
	`[dbms sqlsrvr]`
	`driver=mssdm32.dll`
	`model=tmmss132.dll`

Sybase Version 10 using DB-Library	`[databases]`
	`installed=sybase`
	`[dbms sybase]`
	`driver=db10dm32.dll`
	`model=tmsyb132.dll`

Sybase Version 10 using CT-Library	`[databases]`
	`installed=sybase`
	`[dbms sybase]`
	`driver=ct10dm32.dll`
	`model=tmsyb132.dll`

Sybase Version 11 using DB-Library	`[databases]`
	`installed=sybase`
	`[dbms sybase]`
	`driver=db11dm32.dll`
	`model=tmsyb132.dll`

Sybase Version 11 using CT-Library	`[databases]`
	`installed=sybase`
	`[dbms sybase]`
	`driver=ct11dm32.dll`
	`model=tmsyb132.dll`

For Windows Clients:

Database and Version	Settings
Informix Version 7.20.TDI	[databases] installed=informix [dbms informix] driver=inf7dm32.dll model=tminf132.dll
Oracle Version 8.1.5 using OCI	[databases] installed=oracle [dbms oracle] driver=ora815dm32.dll model=tmora132.dll
Oracle Version 8.1.5 using Pro*C	[databases] installed=oracle [dbms oracle] driver=emb815dm32.dll model=tmora132.dll
Oracle Version 9 using OCI	[databases] installed=oracle [dbms oracle] driver=ora9dm32.dll model=tmora132.dll
Oracle Version 9 using Pro*C	[databases] installed=oracle [dbms oracle] driver=emb9dm32.dll model=tmora132.dll

For Windows Servers:

Database and version	Settings

Informix Version 7.10	`[databases]`
	`installed=informix`
	`[dbms informix]`
	`driver=inf7dmnt.dll`
	`model=tminf132.dll`

Informix Version 7.20TE1 or higher	`[databases]`
	`installed=informix`
	`[dbms informix]`
	`driver=inf7dmn4.dll`
	`model=tminf132.dll`

Oracle Version 8.1.5 using OCI	`[databases]`
	`installed=oracle`
	`[dbms oracle]`
	`driver=ora815dm32.dll`
	`model=tmoral132.dll`

Oracle Version 8.1.5 using Pro*C	`[databases]`
	`installed=oracle`
	`[dbms oracle]`
	`driver=emb815dm32.dll`
	`model=tmora132.dll`

Oracle Version 8.1.5 using OCI with XA sup port	`[databases]`
	`installed=oracle`
	`[dbms oracle]`
	`driver=oxa815dm32.dll`
	`model=tmoral132.dll`

Database and version	Settings
Oracle Version 8.1.5 using Pro*C with XA support	[databases] installed=oracle [dbms oracle] driver=exa815dm32.dll model=tmora132.dll
Oracle Version 9 using OCI	[databases] installed=oracle [dbms oracle] driver=ora9dm32.dll model=tmoral132.dll
Oracle Version 9 using Pro*C	[databases] installed=oracle [dbms oracle] driver=emb9dm32.dll model=tmora132.dll
Oracle Version 9 using OCI with XA support	[databases] installed=oracle [dbms oracle] driver=oxa9dm32.dll model=tmoral132.dll
Oracle Version 9 using Pro*C with XA support	[databases] installed=oracle [dbms oracle] driver=exa9dm32.dll model=tmora132.dll

APPENDIX

E License Administration

This appendix provides background information on licensing in Panther. To manage Panther usage, the Flexible License Manager (FLEX*lm*), a product of Globetrotter Software, Inc., is used. FLEX*lm* Version 5.0 is installed with Panther as part of the installation process only on server machines.

The appendix describes the license options file and the following FLEX*lm* license administration utilities provided with Panther.

- `lmcksum`—Performs a checksum of a license file.
- `lmdiag`—Diagnoses licensing problems.
- `lmdown`—Gracefully shuts down the license daemons on all nodes.
- `lmgrd`—Starts the license manager daemon.
- `lmhostid`—Prints the hostid values on a machine supported by FLEX*lm*.
- `lminstall`—Create and delete Windows service for license manager daemon.
- `lmremove`—Removes a single user's license for a specified feature.
- `lmreread`—Causes the license daemon to reread the license file and start any new vendor daemons that have been added.
- `lmstat`—Monitors the status of all licensing activity.
- `lmswitchr`—Switches the log file for the specified feature.
- `lmver`—Reports the FLEX*lm* version of a library or binary.

Licensing in Panther

Panther components use local or remote licensing schemes and are defined as follows:

- Local licensing is when the license file for the application resides on the same machine as the Panther component being used.
- Remote licensing is when the license file for the application resides on a different machine from the Panther component being used.

The setup program recommends a default licensing scheme for each Panther component. If you accept the default, the setup program guides you through the appropriate steps to obtain a permanent license file. You can choose a different scheme if your license administrator recommends it. In this case, the setup program requests contact information, and the Prolifics License Desk will contact you or your license administrator to arrange for licensing. Default license schemes are:

Component name	Platforms	Licensing scheme
Panther client	UNIX	Local or remote
	Windows	Local or remote
Panther application server engine	UNIX	Local
	Windows	Local
Panther web application broker	UNIX	Local
	Windows	Local

License Daemons and License Types

The license daemons are programs that manage the license types that allow limited access to users. License daemons are only required for node-locked counted and floating licenses. Thus, only the Panther development client requires a daemon; and the license daemon always runs on the license host machine.

The license type and number used at your site was determined when Panther was purchased.

Types of Daemons

The two types of daemons are:

- License daemon (`lmgrd`) — Manages the license file and starts the vendor daemon `prold`. Only one license daemon can be active for a given license file. However, there might be other license daemons managing other license files. `lmgrd` runs on the license host.
- Vendor daemon (`prold`) — Works with `lmgrd` to dispense tokens. Each feature of a floating (development client) or node-locked counted license has a limited number of tokens.

License Types

Two types of licenses are available with Panther; each type is specific to the component installed:

- A Panther application server engine uses a license type that allows for an unlimited number of application server engines to run on a specific computer. The license is does not require a license daemon. This type of license is referred to as *node-locked uncounted*.
- A Panther web application broker uses a node-locked uncounted license which allows for an unlimited number of applications to be run on a specific computer. The license does not require a license daemon.
- Panther clients use *floating* licenses. This type of license is not associated with a particular machine, but is assigned by the license server to each Panther development client when the executable is run. This license type requires a license daemon.

  Note: Because a floating license is associated with the Panther development executable, and not with an actual machine, it is possible for a developer to use more than one license. For example, if a developer is running two Panther sessions simultaneously on one machine, the license manager assigns two licenses, not one.

Options File

The daemon options file allows you to customize Panther license usage at your site. The information in the file can specify such information as:

- Which users or groups of users can use Panther.
- Which messages will be included in the log file.
- How long a copy of Panther will remain idle before timing out.

There is no default location or name for the options file. If used, its name appears as the fourth argument on the `DAEMON` line of the license file `license.dat`, in the optional field *`options-file`*. If there are multiple `DAEMON` lines in the `license.dat` file, then there can be multiple options files, one for each `DAEMON` line. Not all of the lines in an options file refer to a feature, so the site administrator must set up separate options files in order to use the `NOLOG` and `REPORTLOG` features.

Contents of the Options File

The options file has the following basic format:

```
{INCLUDE|EXCLUDE} feature {USER|HOST|DISPLAY|GROUP} name
NOLOG {IN|OUT|DENIED|QUEUED}
GROUP group-name member-list
LINGER feature checkout-time
REPORTLOG filename
RESERVE numlic feature {USER|HOST|DISPLAY|GROUP} name
TIMEOUT feature idletime
```

Lines beginning with a pound sign (#) indicate comments and are ignored.

`INCLUDE/EXCLUDE`

`INCLUDE` and `EXCLUDE` specify which users (or hosts, displays, or groups) are allowed to use a particular feature. Any user who is `EXCLUDE`d from a feature is not able to use that feature. Specifying an `INCLUDE` line has the effect of excluding everyone else from that feature; thus, only those users specifically

INCLUDEd are able to use that feature. The INCLUDE|EXCLUDE line has the following format:

```
{INCLUDE|EXCLUDE} feature {USER|HOST|DISPLAY|GROUP} name
```

NOLOG

NOLOG causes messages of the specified type to be filtered out of the daemon's log file. Specifying a NOLOG option reduces the amount of output to the log file, which can be useful in those cases where the log file grows too quickly. The NOLOG line has the following format:

```
NOLOG {IN|OUT|DENIED|QUEUED}
```

GROUP

GROUP defines collections of users, which can then be used in RESERVE, INCLUDE, or EXCLUDE commands. The GROUP line has the following format:

```
GROUP group-name member-list
```

LINGER

LINGER prevents the license manager from taking back a license until the indicated *checkout-time* has expired, regardless of whether or not the license token is being used. The LINGER line has the following format:

```
LINGER feature checkout-time
```

REPORTLOG

REPORTLOG creates a log file suitable for use with the FLEX*lm* report writing tools. This log file maintains more detailed information than the standard log file, but is not meant to be human readable. If the filename starts with a plus character (+), the file will be opened in append mode. The REPORTLOG line has the following format:

```
REPORTLOG filename
```

RESERVE

RESERVE reserves the specified number of licenses for the specified user, host, display, or group. Reserving a license decreases the number of generally available licenses. The RESERVE line has the following format:

```
RESERVE numlic feature {USER|HOST|DISPLAY|GROUP} name
```

TIMEOUT

TIMEOUT sets up a minimum idle time after which a user's license is lost if it is not being used. This can prevent users from wasting a license (by keeping it checked out when it is not in use) when someone else wants one. The TIMEOUT line has the following format:

```
TIMEOUT feature idletime
```

Example The following is an example of an options file:

```
REPORTLOG /usr/adm/gsi.replog
RESERVE compile USER pat
RESERVE compile USER less
RESERVE compile HOST terry
NOLOG QUEUED
```

FLEX*lm* Utilities

The following sections describe the FLEX*lm* utility programs provided with Panther. These utilities are located in the distributed `util` directory.

lmcksum

Performs a checksum of a license file

`lmcksum [-k] [-c license_file]`

`-k`

Forces the encryption code checksum to be case-sensitive (in general, `lmcksum` is not case-sensitive) and not prompt for any input.

`-c license_file`

Name of license file to checksum. By default `lmcksum` operates on `license.dat` in the current directory. If this switch is not specified, `lmcksum` looks for the environment variable `LM_LICENSE_FILE`. If the environment variable is not set, `lmdiag` looks for the file `/usr/local/flexlm/licenses/license.dat`.

Description

`lmcksum` prints a line-by-line checksum for the file as well as an overall file checksum. If the license file contains "`lmcksum=nn`" attributes, the bad lines are indicated.

`lmcksum` ignores all fields that do not enter into the encryption code computation; thus the server node name and port number, as well as the daemon pathname and options file names are not included in the checksum. In addition `lmcksum` treats non-case sensitive fields correctly (in general, `lmcksum` is not case-sensitive). `lmcksum` takes the `-k` switch to force the encryption code checksum to be case-sensitive.

`lmcksum` takes an optional daemon name; if a name is specified, only license file lines for the selected daemon are used to compute the checksums.

lmdiag

Diagnoses licensing problems

`lmdiag [-c license_file] [-n] [feature]`

`-c license_file`
: Name of license file to diagnose. If this switch is not specified, `lmdiag` looks for the environment variable `LM_LICENSE_FILE`. If the environment variable is not set, `lmdiag` looks for the file `/usr/local/flexlm/licenses/license.dat`.

`-n`
: Non-interactive mode; `lmdiag` does not prompt for any input. In this mode, extended connection diagnostics are not available.

`feature`
: Diagnose only the specified *`feature`*.

Description

If no *`feature`* is specified, `lmdiag` operates on all features in the license file in your path. `lmdiag` first prints information about the license, then attempts to check out each license. If the checkout succeeds, `lmdiag` indicates this. If the checkout fails, `lmdiag` gives you the reason for the failure. If the checkout fails because `lmdiag` cannot connect to the license server, then you have the option of running "extended connection diagnostics."

Extended diagnostics attempt to connect to each port on the license server node, and can detect if the port number in the license file is incorrect. `lmdiag` indicates each port number that is listening, and if it is an `lmgrd` process, `lmdiag` indicates this as well. If `lmdiag` finds the vendor daemon for the *`feature`* being tested, then it indicates the correct port number for the license file to correct the problem.

lmdown

Takes down license daemons

```
lmdown [-c license_file] [-q]
```

-c *license_file*
: Use the specified *license_file*. If this switch is not specified, `lmdown` looks for the environment variable `LM_LICENSE_FILE`. If the environment variable is not set, `lmdown` looks for the file `/usr/local/flexlm/licenses/license.dat`.

`-q`
: Quiet mode; `lmdown` does not ask for confirmation. If the switch is not specified, `lmdown` asks for confirmation before asking the license daemons to shut down.

Description

`lmdown` sends a message to every license daemon asking it to shut down. The license daemons write out their last messages to the log file, close the file, and exit. All licenses which have been given out by those daemons are rescinded, so that the next time a client program goes to verify its license, it will not be valid.

The end-user system administrator should protect the execution of `lmdown` since shutting down the servers causes loss of licenses.

Note: `lmdown` can be used only by a "FLEX*lm* administrator" (i.e., a member of group `lmadmin` or, if the `lmadmin` group does not exist, a member of group `0`).

lmgrd

Starts up the license manager daemon

```
lmgrd [-2] [-b] [-c license_file] [-d] [-l logfile] [-p]
   [-s interval] [-t timeout]
```

`-2`
: Specifies V2 startup arguments, in contrast to the `-b` switch. This switch is required if you intend to use the `-p` switch (available in `lmgrd` v2.4 and later).

`-b`
: Specifies backward compatibility mode. Use this switch if you are running a v2.1 or later `lmgrd` with a v1.5 or earlier vendor daemon. This is the default switch in FLEX*lm* v2.4 and later.

`-c` *`license_file`*
: Use the specified *`license_file`*. If this switch is not specified, `lmgrd` looks for the environment variable `LM_LICENSE_FILE`. If the environment variable is not set, `lmgrd` looks for the file `/usr/local/flexlm/licenses/license.dat`.

`-d`
: Specifies that hostnames which are read from the license file should have the local domain name appended to them before sending to a client. Useful when clients are accessing licenses from another domain. (Available in `lmgrd` v2.4 and later.)

`-l` *`logfile`*
: Specifies the output log file to use.

`-p`
: Specifies that the `lmdown` and `lmremove` utilities can only be run by a license administrator. A license administrator is a member of the `lmadmin` group, or, if the `lmadmin` group does not exist, a member of group `0`. (This is available in `lmgrd` v2.4 and later.)

`-s` *`interval`*
: Specifies the logfile timestamp interval, in minutes. Default is 360 minutes.

`-t` *`timeout`*
: Specifies the timeout interval, in seconds, during which daemons must complete their connections to each other. Default value is 10 seconds. A

larger value might be preferable if the daemons are being run on busy systems or a very heavily loaded network.

Environment UNIX

Description `lmgrd` is the main daemon program for the FLEX*lm* distributed license management system. When invoked, it looks for a license file containing all required information about vendors and features.

lmhostid

Prints the correct hostid value on any machine supported by FLEXlm

`lmhostid [`*`type`*`]`

`type`

The type of the hostid to print. *`type`* must be one of `long`, `idmodule`, `ether`, or `string`, and is currently used only on HP and SCO systems. On HP, *`type`* specifies the ID module, the machine id as returned from the `uname` command, or the Ethernet address. The HP default is `long-uname`. On SCO, `long` specifies the pre-3.0 default, which was a 32-bit long int, while `string` specifies a string host, which is the new default.

Description

Output from `lmhostid` is similar to the following:

```
lmhostid - Copyright (C) 1989-1999 Globetrotter Software, Inc.
The FLEXlm host ID of this machine is "1700abcd"
```

lminstall

Create and delete Windows service for license manager daemon

```
lminstall [-n Service_Name] [-c license_file] [-l <log_file>]
    -e <path_to_lmgrd>

lminstall -r [-n Service_Name]
```

`--n Service_Name`
: Specifies the name of the Windows service to create or delete. If omitted, `"FLEXlm License Server"` is used as service name.

`-c license_file`
: Use the specified license file. If this switch is not specified, `lminstall` looks for the environment variable `LM_LICENSE_FILE`. If the environment variable is not set, it looks for the file `/usr/local/flexlm/licenses/license.dat`.

`-l <log_file>`
: Path to the log file to use for recording license activity.

`-e <path_to_lmgrd>`
: Path to `lmgrd.exe`.

`-r`
: Remove the service.

Description

`lminstall` allows a system administrator to create or remove a Windows service that runs the license manager daemon `lmgrd.exe`.

lmremove

Removes a user license and returns it to the license pool

```
lmremove [-c license_file] [feature] [user] [host] [display]
```

-c *license_file*
: Use the specified license file. If this switch is not specified, `lmremove` looks for the environment variable `LM_LICENSE_FILE`. If the environment variable is not set, it looks for the file `/usr/local/flexlm/licenses/license.dat`.

feature
: Remove only from the specified *feature*.

user
: The *user* to be removed.

host
: Node from which the *user* is to be removed.

Description

`lmremove` allows the system administrator to remove a single user's license for a specified *feature*. This might be required in the case where the licensed user is running the software on a node that subsequently crashed. This situation can some times cause the license to remain unusable. `lmremove` allows the license to be re turned to the pool of available licenses.

`lmremove` removes all instances of *user* on node *host* at the specified *display* from usage of *feature*. The end-user system administrator should protect the execution of `lmremove` since removing a user's license can be disruptive.

Note: `lmremove` can be used only by a "FLEX*lm* administrator" (i.e., a member of group `lmadmin` or, if the `lmadmin` group does not exist, a member of group `0`).

lmreread

Tells the license daemon to reread the license file

`lmreread [-c license_file]`

`-c license_file`
: Use the specified license file. If this switch is not specified, `lmreread` looks for the environment variable `LM_LICENSE_FILE`. If the environment variable is not set, it looks for the file `/usr/local/flexlm/licenses/license.dat`.

Description

`lmreread` allows the system administrator to tell the license daemon to reread the license file. This can be useful if the data in the license file has changed; the new data can be loaded into the license daemon without shutting down and restarting it.

`lmreread` uses the license file from the command line (or the default file, if none specified) only to find the license daemon to send it the command to reread the license file. The license daemon always rereads the original file that it loaded. If you need to change the path to the license file, then you must shut down the daemon and restart it with that new license file path.

You cannot use `lmreread` if the `SERVER` node names or port numbers have been changed in the license file. In this case, you must shut down the daemon and restart it in order for those changes to take effect.

`lmreread` does not change any option information specified in an options file. If the new license file specifies a different options file, that information is ignored. If you need to reread the options file, you must shut down the daemon and restart it.

lmstat

Reports status on license manager daemons and feature usage

`lmstat [-a] [-A] [-c license_file] [-f [feature]] [-l [reg_expression]] [-s[server]] [-S[daemon]] [-t timeout]`

-a
: Display everything.

-A
: List all active licenses.

-c *license_file*
: Use the specified license file. If this switch is not specified, `lmstat` looks for the environment variable `LM_LICENSE_FILE`. If the environment variable is not set, `lmstat` looks for the file `/usr/local/flexlm/licenses/license.dat`.

-f [*feature*]
: List all users of the specified features.

-l [*reg_expression*]
: List all users of the features matching the given regular expression.

-s [*server*]
: Display the status of the specified server nodes.

-S[*daemon*]
: List all users of the specified daemon's features.

`-t` *timeout*
: Specifies the timeout interval, in seconds, during which daemons must complete their connections to each other. Default value is 10 seconds. A larger value might be desirable if the daemons are being run on busy systems or a very heavily loaded network.

Description `lmstat` provides information about the status of the server nodes, vendor daemons, vendor features, and users of each feature. Information can optionally be qualified by specific server nodes, vendor daemons, or features.

`lmstat` provides:

- Total licenses available for each feature.
- Who is using the license.
- Who is using features served by a specific daemon.

lmswitchr

Switches the FLEXadmin log file (REPORTLOG) for the specified feature

`lmswitchr [feature] [new-file]`

feature
: Any *feature* this daemon supports.

new-file
: New file pathname.

Description

The FLEX*lm* daemons create an ASCII log file on `stdout`. There are several processes in a parent-child hierarchy which share the same file pointer, so this log file cannot be changed after the vendor daemons have been started, since each process has a copy of the current offset, etc.

There is another way to switch the log file output data; however, this involves piping the `stdout` of `lmgrd` to a shell script that appends a file for each line. Instead of the "normal" startup: `% lmgrd > LOG`

Start `lmgrd` this way:

```
% lmgrd -z | sh -c 'while read line; do echo "$line" >> LOG; done'
```

For more information on the log file, refer to page E-5, "REPORTLOG".

lmver

Reports the FLEXlm version of a library or binary

`lmver [filename]`

filename
: Use the specified license file. If a filename is not specified, `lmver` looks for the environment variable `LM_LICENSE_FILE`. If the environment variable is not set, `lmver` looks for the file `/usr/local/flexlm/licenses/license.dat`.

Description

If the filename is specified, the FLEX*lm* version incorporated into this file is displayed; otherwise `lmver` looks for the library file `liblmgr.a` to detect its version.

APPENDIX

F License File

This appendix describes the basic format and content of the password-enabled license file.

If you already have a license file in place, either from a previous version of JAM or Prolifics, or from another application that uses FLEX*lm*, you can combine license files by inserting the license file information provided by the License Desk.

Contents of the License File

The license file has the following basic format:

```
SERVER sname hostid port

DAEMON daemon-name path

FEATURE product daemon-name version exp_date number password \
    ISSUED=date SN=serial_number HOSTID=hostid ck=checksum

INCREMENT product daemon-name version exp_date number password \
    ISSUED=date SN=serial_number HOSTID=hostid ck=checksum
```

Note: `FEATURE` and `INCREMENT` lines should not be changed: they contain the license password and other information which, if modified, could prevent Panther from running. Also, the host ID of the server on the `SERVER` line should not be changed.

SERVER

Specifies the server on which a license manager can run. License files require a SERVER line only for the Panther Editor. The SERVER line has the following format:

SERVER *sname hostid port*

sname — Server's name.

hostid — Host ID of the server.

port — TCP/IP port number used by this server.

DAEMON

Specifies the name and location of the license manager vendor daemon for the Panther development client. It also provides the location of the Panther options file. It has the following format:

DAEMON *daemon-name path* [*options_file*]

daemon-name — Name of the daemon; always prold.

path — Full path for the daemon.

options_file (optional) — Name and path of the options file allowing you to customize Panther license management (for more information, refer to page E-4, "Options File."

FEATURE

Describes the license for the Panther application server engine or the web application broker. It has the following format:

```
FEATURE product daemon-name version exp_date number \
    password ISSUED=date SN=serial_number \
    HOSTID=hostid ck=checksum
```

product — Name of the product with which the license is associated, in this case, the appropriate Panther executable:

- prolifics-client (Panther client)
- prolifics-ejb-server (Panther application server engine for IBM Web Sphere)
- prolifics-server (Panther application server engine for JetNet and Oracle Tuxedo)
- prolifics-web-server (Panther web application broker or jserver)
- prolifics-web-mgr (dispatcher)

daemon-name — Always no_daemon; this indicates that the feature requires no license daemons.

version — Highest version number of *product*.

exp_date — Expiration date of the license in the form dd-mmm-yyyy; normally 01-jan-0000 to indicate that the Panther license has no expiration date.

number — Number of licenses allowed under this license agreement: it is always 0 for a node-locked uncounted license.

password — Encrypted alphanumeric string provided by the License Desk. It contains encoded information about your license, such as the type of license and your system configuration.

date — Date issued. The license is not valid before this date.

serial_number — Product serial number; more than one feature can share a serial number.

hostid — Hostid of the server. Used only if the feature is to be bound to a particular host, whether its use is counted or not.

checksum — Verifies that the license has been entered correctly by the end-user.

INCREMENT

Describes the license for the Panther development client. It has the following format:

```
INCREMENT product daemon-name version exp_date number \
    password ISSUED=date SN=serial_number \
    HOSTID=hostid ck=checksum
```

product — Name of the product with which the license is associated, in this case, the appropriate Panther executable: prolifics-client (UNIX or Windows clients).

daemon-name — Name of the daemon; always prold.

version — Highest version number of *product*.

exp_date — Expiration date of the license in the form dd-mmm-yyyy; normally 01-jan-0000 to indicate that the Prolifics license has no expiration date.

number — Number of licenses allowed under this license agreement: it is always greater than 0 for a counted license.

password — Encrypted alphanumeric string provided by the License Desk. It contains encoded information about your license, such as the type of license and your system configuration.

date — Date issued. The license is not valid before this date.

serial_number — Product serial number; more than one feature can share a serial number.

hostid — Hostid of the server. Required for a Panther web application server and absent for a development client.

checksum — Verifies that the license has been entered correctly by the end-user.

APPENDIX

G Error Messages

Panther Installation Error Messages

A Panther installation was not successful if you receive an error message and the program terminates. The following messages describe some of the more common error messages and how to resolve them.

Recurring errors: Occasionally, errors described in this section seem to recur even though the file `smvars` is in order. More often than not, the problem is that the `smvars` file (in the `config` directory), although correct, has not been converted to binary (`smvars.bin`), which is the required format used by Panther. If the ASCII version of `smvars` has been edited, be sure to run `var2bin` to convert it to binary format. Otherwise Panther is not aware of the changes, and it will seem as though `smvars` was not updated.

```
Please enter terminal type or <RETURN> to exit.
```

Cause:

`SMTERM` is not set (and, under UNIX, `TERM` is also not set) or the value could not be found in `smvars.bin` (left column of `smvars`).

Action:

Enter the terminal type if known or press Enter to set `SMTERM` correctly. Once the variable is defined, run `prodev` again.

See the left column of the `smvars` file and correct the specification. Use `var2bin` to convert the file to binary. And/Or:

UNIX: Set the variable at the command line.

Windows: Set the variable to `mswin` in `prol5w32.ini` or `prol5w64.ini`.

If you are running in character mode and the screen appears disorganized, with all the text bunched together along with special characters, check the value of `SMTERM`; you probably only need to change its value, but you may also need to create a new video file.

`Filename: No such file or directory`

Cause:

The filename in brackets is specified as the value of the `SMVIDEO`, `SMKEY`, `SMMSGS`, or some similar configuration variable in the environment or in `smvars.bin`, but the file could not be found.

Action:

Correct the name, and use the full pathname of the file. If you correct it in `smvars`, recompile `smvars` with the `var2bin` utility.

`SMMSGS``: Environment variable missing`

Cause:

(or the same message with some other configuration variable). Panther could not find an entry for the cited variable in `SMVARS` or in the environment (refer to the following note). This error usually means that you have some variables defined in your environment, but not all that are required, and `SMVARS` is not defined.

Note: Most Panther setup variables can be set in the environment, rather than depending on values set in an `SMVARS`-defined file. If Panther can't find a required variable specification in either the environment or in the `SMVARS` file, that variable will be cited as missing. Refer to the *Configuration Guide* for more details.

`SMVARS``: Bad file format`

Cause:

`SMVARS` is set to be the name of an existing file, but it is not a binary `SMVARS` file. The most common incorrect value for `SMVARS` in this case is the path and name of the source `SMVARS` file instead of the binary `SMVARS` file.

Action:

Point to the binary file that has the `.bin` extension.

SMVARS: `Environment variable missing`

Cause:

`SMBASE` (or `SMVARS`, if you are using one) is not set, or not set properly. Normally, Panther looks for `smvars.bin` in the `config` directory under the directory pointed to by `SMBASE`. If `SMBASE` is not set, you must have an `SMVARS` variable to tell Panther explicitly where the `SMVARS` file can be found. If `SMBASE` is not set property, and Panther cannot find an `SMVARS` variable, the above message is given.

Action:

Correct (or set) the `SMBASE` variable specification, or determine why setting it is not taking effect.

Windows: Check to see that `SMBASE` is being set in `autoexec.bat` and that you have not run out of environment space.

UNIX: Make sure that the variable is exported (via the `export sh/ksh` command or `setenv csh` command).

SMVARS: `No such file or directory`

Cause:

`SMVARS` is not set correctly.

Action:

Check that the full pathname (including drive letter under Windows) of the file is included and correct.

DLL Messages

The following error messages might occur after a Windows installation and are specific to DLLs required for Panther's database drivers and to DLL version mismatches.

Oracle Tuxedo-Specific Messages

```
Cannot find LIBWSC.DLL
```

Cause:

Windows is unable to find the Oracle TUXEDO DLLs.

Action:

Check that Oracle Tuxedo is present and configured properly.

Informix-Specific Messages

```
Cannot load DLL INF7DM32.DLL
Cannot load DLL INF7DMNT.DLL
Cannot load DLL INF7DMN4.DLL
Cannot load DLL INF9DMN4.DLL
Cannot load DLL TMINF132.DLL
```

Cause:

Windows is unable to find the Panther database driver DLLs or the Informix software.

Action:

Check that the Panther `util` directory is on the `PATH`. If you do not wish to use the Panther database driver DLLs, edit `PROL5W32.INI` to remove the database name from the `install` entry.

Microsoft SQL Server-Specific Messages

```
Cannot load DLL mssdm32.dll
Cannot load DLL tmmss132.dll
```

Cause:

Windows is unable to find the Panther database driver DLLs or the Microsoft SQL Server software.

Action:

Check that the Panther `util` directory is on the `PATH`. If you do not wish to use the Panther database driver DLLs, edit `PROL5W32.INI` to remove the database name from the `install` entry.

ODBC-Specific Messages

```
Cannot load DLL odb2dm32.dll
Cannot load DLL odb3dm32.dll
Cannot load DLL tmodb132.dll
```

Cause:

Windows is unable to find the Panther database driver DLLs or the ODBC software.

Action:

Check that the Panther `util` directory is on the `PATH`. If you do not wish to use the Panther database driver DLLs, edit `PROL5W32.INI` to remove the database name from the install entry.

Oracle-Specific Messages

```
Cannot load DLL ora815dm32.dll
Cannot load DLL emb815dm32.dll
Cannot load DLL oxa815dm32.dll
Cannot load DLL exa815dm32.dll

Cannot load DLL ora9dm32.dll
Cannot load DLL emb9dm32.dll
Cannot load DLL oxa9dm32.dll
Cannot load DLL exa89m32.dll

Cannot load DLL tmora132.dll
```

Cause:

Windows is unable to find the Panther database driver DLLs or the Oracle software.

Action:

Check that the Panther `util` directory is on the `PATH`. Verify that Oracle is installed correctly. If you do not wish to use the Panther database driver DLLs, edit `PROL5W32.INI` to remove the database name from the `install` entry.

Sybase-Specific Messages

```
Cannot load DLL db10dm32.dll
Cannot load DLL ct10dm32.dll
Cannot load DLL db11dm32.dll
Cannot load DLL ct11dm32.dll
Cannot load DLL tmsyb132.dll
```

Cause:

Windows is unable to find the Panther database driver DLLs or the Sybase software.

Action:

Check that the Panther `util` directory is on the `PATH`. If you do not wish to use the Panther database driver DLLs, edit `PROL5W32.INI` to remove the database name from the `install` entry.

DLL Version Mismatch Messages

`Application Error: Call to Undefined Dynalink`

Cause:

Panther is picking up an older version of `cktbl32.dll` from a previous installation.

Action:

Check that the current Panther `util` directory with the correct `cktbl32.dll` is on the `PATH` or copy `cktbl32.dll` to the Windows directory.

Graph-Related Messages

If graph widgets are not displaying correctly, the following graph-specific errors might be displayed;

`Warning: chart <Begin> failed: -1`

Cause:

All graphs are blank indicating that the `gdsp` program (in the `util` directory) was not found on the path. (UNIX only.)

`Warning: missing graph files in $SMPATH`

Cause:

All graphs are blank indicating that: the `grafcap` file is invalid or not found in `SMPATH` (UNIX) and the IPT setting in the file `LIBSTI.INI` does not point to the directory that contains the `grafcap` file (Windows).

License Manager Error Messages

In most circumstances, Panther license management is transparent to the developer. This section describes the error messages that can be generated from Panther and from the underlying license manager software, FLEX*lm*.

Panther License-Related Messages

The messages described in this section can appear when the screen editor is invoked.

All License Types

```
Prolifics License Manager: Bad date in license file.
```

Cause:

The date on one of the `FEATURE` or `INCREMENT` lines is not correct.

Action:

Verify that the `FEATURE` and `INCREMENT` lines in the license file are correct or contact your system administrator.

```
Prolifics License Manager: Bad encryption code in license file.
```

Cause:

The password on one of the `FEATURE` or `INCREMENT` lines is not correct.

Action:

Verify that the `FEATURE` and `INCREMENT` lines in the license file are correct and that none of the backspaces indicating continued lines are followed by a space, or contact your system administrator.

`Prolifics License Manager: Cannot communicate with server. cannot connect to license server ([code,code])`

Cause:

Could not connect with the license server.

Action:

Make sure the license server daemon `lmgrd` is started on the system specified in the `SERVER` line of the license file, or contact your system administrator. The two codes can help locate the problem.

`Prolifics License Manager: Cannot connect daemon prold with license server.`

Cause:

There is no `DAEMON` line in the license file for `prold`.

Action:

Verify that the license file entry is correct or contact your system administrator.

`Prolifics License Manager: Cannot find license file.`

Cause:

The license manager is unable to find the license file.

Action:

Make sure `LM_LICENSE_FILE` in your initialization file or environment points to the correct license file and that it is readable.

`Prolifics License Manager: Cannot read license file`

Cause:

The license manager could not read the license file because of a permissions or access problem.

Action:

Correct the problem or contact your system administrator.

`Prolifics License Manager: ERROR #[code] , [message]`

Cause:

An unexpected error occurred in the license manager.

Action:

Contact your system administrator.

```
Prolifics License Manager: [product] feature does not exist.
```

Cause:

There is no FEATURE or INCREMENT line in the license file for [product].

Action:

Verify that the FEATURE or INCREMENT lines in the license file are correct, obtain a new license file, or contact your system administrator.

```
Prolifics License Manager: Feature [product], Version [version] has expired.
```

Cause:

The expiration date in the license file of [product] has been reached.

Action:

Obtain a new license file or contact your system administrator.

```
Prolifics License Manager: Initialization failed.
```

Cause:

Problems in the license file.

Action:

Contact your system administrator.

```
Prolifics License Manager: Invalid data received from license server.
```

Cause:

The license server did not respond to a request or the response was invalid.

Action:

Contact your system administrator.

```
Prolifics License Manager: Invalid license file syntax.
```

Cause:

A feature name, daemon name, or server name is too long or a FEATURE or INCREMENT line specifies no hostid, or the number of licenses is not greater than zero.

Action:

Verify that the license file entry is correct or contact your system administrator.

```
Prolifics License Manager: malloc() call failed
```

Cause:

License manager could not allocate memory that it needed. The most likely cause is that the program's heap has been corrupted.

Action:

Contact your system administrator.

```
Prolifics License Manager: Network software (TCP/IP) not available.
```

Cause:

Could not communicate with the license manager because network software is not present or is correctly configured.

Action:

Contact your system administrator.

```
Prolifics License Manager: Platform not enabled.
```

Cause:

Should not happen. Indicates that license daemon `prold` is not licensed to run on your platform.

Action:

Contact your system administrator.

```
Prolifics License Manager: Server does not support feature [product].
```

Cause:

The feature [product] does not have a `FEATURE` or `INCREMENT` line in the license file.

Action:

Verify that the license file entry is correct, obtain a new license file, or contact your system administrator.

```
Prolifics License Manager: Time zone offset from GMT more then 24 hours
```

Action:

Fix the TZ environment variable or contact your system administrator.

```
Prolifics License Manager: Unknown vendor key type
```

Cause:

Program's object file is corrupt.

Action:

Contact your system administrator.

Floating Licenses

```
Prolifics License Manager: All [product] licenses are currently in use.
```

Cause:

All available licenses for [product] are in use. If your license request has been queued, you get a series of messages showing who has the licenses reserved or how many licenses are in use.

Action:

Try again later or contact your system administrator.

```
Prolifics License Manager: Cannot communicate with license server or your
[product] license was obtained by another user when the license server was
restarted.
```

Cause:

Occurs when you select an item from the New or Open menu options; indicates that the license server has been taken down. If the license server was then restarted, another user got the license for the [product] you were using.

Action:

You can continue editing and can save any screens that are currently open in the editor but must restart Panther (`prodev`) before you can pick any New and Open menu options.

```
Prolifics License Manager: FLEXlm key data bad.
```

Cause:

Program's object file is corrupt.

Action:

Contact your system administrator.

```
Prolifics License Manager: Bad server hostname in license file.
```

Cause:

Server named on a `SERVER` line could not be found.

Action:

Obtain a new license file or contact your system administrator.

```
Prolifics License Manager: No SERVER lines in license file.
```

Cause:

The license file cannot be used because there are no `SERVER` lines for a counted feature.

Action:

Verify that the license file entry is correct, obtain a new license file, or contact your system administrator.

```
Prolifics License Manager: Version [version] not supported on server.
```

Cause:

The version [version] of the program being run is greater than that supported by the license file.

Action:

Obtain a new license file or contact your system administrator.

Node-locked Licenses

The following Panther startup error information is specific to node-locked licenses.

```
Prolifics License Manager: Cannot find ethernet device.
```

Cause:

On most systems, the license manager must locate an ethernet board to verify the password on the `FEATURE` or `INCREMENT` lines.

Action:

Contact your system administrator.

`Prolifics License Manager: [product] not authorized for this computer.`

Cause:

[*product*] is not licensed to run on the current computer.

Action:

Contact your system administrator.

`Prolifics License Manager: Version [version] not supported.`

Cause:

The version [*version*] of the program being run is greater than that.

Action:

Obtain a new license file or contact your system administrator.

Create License Utility

The following error information is specific to the `create_license` utility.

`No license files (.lic) were detected.`

Cause:

Could not detect any license files (`.lic`) in the current directory.

Action:

Check the filenames of the license files in `$SMBASE/licenses`. Your license files should have one of the following names: `proserv.lic` (Panther application server engine), `prodev.lic` (development client), or `proweb.lic` (web application broker).

`Unable to create license.dat`

Cause:

Encountered an error when trying to create the `license.dat` file.

Action:

Check file permissions and disk space to allow the `create_license` utility to write to disk.

`Inconsistent DAEMON lines.`

Cause:

Two license files (`.lic`) have `DAEMON` lines, but they are inconsistent with each other. Since the resulting `license.dat` file and the vendor daemon (`prold`) are for a single machine, any `DAEMON` lines among license files should be the same.

Action:

Check the license files with `DAEMON` lines for typographical errors.

`Inconsistent SERVER lines.`

Cause:

Two license files (`.lic`) have `SERVER` lines, but they are inconsistent with each other. Since the resulting license.dat file is meant for one server machine, any `SERVER` lines among license files should be the same.

Action:

Check the license files with `SERVER` lines for typographical errors.

FLEX*lm* Informational Messages

The following messages are generated by FLEX*lm* license management software, and is provided courtesy of the *FLEXlm Programmer's Guide*.

`Connected to` *`node`*

Cause:

This daemon is connected to its peer on *`node`*.

`CONNECTED, master is` *`name`*

Cause:

License daemon logs this message when a quorum is achieved and everyone has selected a master.

`DENIED:` *`N feature`* `to` *`user`* (*`mm/dd/yy hh:mm`*)

Cause:

`user` was denied access to *`N`* licenses of *`feature`*.

`EXITING DUE TO SIGNAL nnn`

`EXITING WITH CODE nnn`

Cause:

An interrupt signal has been intercepted. All daemons list the reason that the daemon has exited.

`EXPIRED: feature`

Cause:

`feature` has passed its expiration date.

`IN: feature` by `user` (`N` licenses) (used: `d:hh:mm:ss`)

Cause:

`user` at d`:hh:mm:ss`.

`IN server died: feature` by `user` (`N` licenses) (used: `d:hh:mm:ss`)

Cause:

`user` has checked in `N` licenses of `feature` by virtue of the fact that his server died.

`License Manager server started`

Cause:

License daemon has been started.

`Lost connection to host`

Cause:

A daemon can no longer communicate with its peer on node `host`, which can cause the clients to have to reconnect, or cause the number of daemons to go below the minimum number, in which case clients may start exiting. If license daemons lose the connection to the master, they will kill all vendor daemons; vendor daemons will shut themselves down.

`Lost quorum`

Cause:

Not enough servers to satisfy quorum number. The daemon will process only connection requests from other daemons.

Action:

Check network connection between servers.

`MASTER SERVER died due to signal` *`nnn`*

Cause:

License daemon received fatal signal *`nnn`*.

`MULTIPLE` *`xxx`* `servers running.`
`Please kill, and restart license daemon`

Cause:

License daemon has detected multiple copies of vendor daemon *`xxx`* are running.

Action:

Kill all *`xxx`* daemon processes and restart the license daemon.

`OUT:` *`feature`* by *`user`* (*`N`* `licenses) (used:` *`d:hh:mm:ss`*)

Cause:

`user` has checked out *`N`* licenses of *`feature`* at *`d:hh:mm:ss`*.

`Removing clients of children`

Cause:

Top-level daemon logs this message when one of the child daemons dies.

`RESERVE` *`feature`* `for HOST` *`name`*
`RESERVE` *`feature`* `for USER` *`name`*

Cause:

A license of *`feature`* is reserved for either user *`name`* or host *`name`*.

Action:

None.

`Restarted` *`xxx`* `(internet port` *`nnn`*`)`

Cause:

Vendor daemon *`xxx`* was restarted at internet port *`nnn`*.

`Retrying socket bind (address in use)`

Cause:

The license server tries to bind sockets for approximately 6 minutes if they detect `address` in use errors.

Action:

Check for multiple `lmgrd`s using the same TCP port number.

`Selected (EXISTING) master` *`node`*

Cause:

This license daemon has selected an existing master (*`node`*) as the master.

`SERVER shutdown requested`

Cause:

Daemon received shutdown request from a user-generated kill command.

`[NEW] Server started for:` *`feature-list`*

Cause:

A (possibly new) server was started for the features listed.

`Shutting down` *`xxx`*

Cause:

The license daemon is shutting down the vendor daemon *`xxx`*.

`SIGCHLD received. Killing child servers.`

Cause:

Vendor daemon logs this message when a shutdown is requested by the license daemon.

`Started` *name*

Cause:

License daemon logs this message whenever it starts a new vendor daemon.

`Trying connection to` *node*

Cause:

Daemon is attempting a connection to *node*.

FLEXlm Configuration Problem Messages

hostname: `Not a valid server host, exiting`

Cause:

This daemon was run on an invalid hostname.

Action:

Run `lmgrd` on the host(s) specified in the `SERVER` lines.

hostname: `Wrong hostid, exiting`

Cause:

Hostid is wrong for *hostname*.

Action:

Check the license file and ensure the hostnames match the hostids.

`BAD CODE for` *feature-name*

Cause:

Specified feature name has a bad encryption code.

Action:

Check the password received from Prolifics.

`CANNOT OPEN options file` *file*

Cause:

Options file specified in the license file could not be opened.

Action:

Check the path for the options file on the `DAEMON` line in the license file.

`Couldn't find a master`

Cause:

Daemons could not agree on a master.

Action:

Kill and then restart `lmgrd` on servers.

`license daemon: lost all connections`

Cause:

Indicates all the connections to a server are lost, which often indicates a network problem.

Action:

Check the network and restart the daemons.

`lm_server: lost all connections`

Cause:

All the connections to a server are lost; probably indicates a network problem.

`lost lock, exiting`

Cause:

Error closing lock file.

`NO DAEMON line for` *`daemon`*

Cause:

License file does not contain a `DAEMON` line for *`daemon`*.

Action:

Add `DAEMON` line for *`daemon`* in the license file.

`NO DAEMON line for` *`name`*

Cause:

Vendor daemon logs this error if it cannot find its own `DAEMON` name in the license file.

Action:

Edit license file.

`NO DAEMON lines, exiting`

Cause:

License daemon logs this message if there are no `DAEMON` lines in the license file; with no vendor daemons to start, there is nothing to do.

Action:

Edit license file.

`No features to serve!`

Cause:

Vendor daemon found no features to serve. This could be caused by bad data in the license file.

Action:

Inspect the license file for bad data.

`No license data for feature, feature unsupported`

Cause:

No feature line for *`feature`* in the license file.

Action:

Edit the license file.

`Unable to re-open lock file`

Cause:

Vendor daemon has a problem with its lock file, usually because of an attempt to run more than one copy of the daemon on a single node.

Action:

Locate the other daemon that is running via a `ps` command, and kill it with `kill -9`.

`Unknown host: hostname`

Cause:

`hostname` specified on a `SERVER` line in the license file does not exist in your host's database.

Action:

Check with your system administrator for the correct hostname. FLEX*lm* uses standard network services to find the host: Domain Name Server (DNS), Network Information Services (NIS or YP) or in `/etc/hosts`.

`UNSUPPORTED FEATURE request: feature by user`

Cause:

`user` has requested a feature that this vendor daemon does not support. This can happen for a number of reasons: the license file is bad, the feature has expired, or the daemon is accessing the wrong license file.

Daemon Software Error Messages

`ATTEMPT TO START VENDOR DAEMON xxx with NO MASTER`

Cause:

Vendor daemon was started with no master selected. This is an internal consistency error in the daemons.

Action:

Report error to Prolifics technical support.

`BAD PID message from nnn: xxx (msg)`

Cause:

Top-level vendor daemon received an invalid PID message from one of its children (daemon number *xxx*).

`BAD SCONNECT message: (`*`message`*`)`

Cause:

An invalid `server` connect message was received.

`Cannot create pipes for server communication`

Cause:

The `pipe` system call failed.

Action:

Report error to Prolifics technical support.

`Can't allocate server table space`

Cause:

A malloc error.

Action:

Check swap space

`Connection to node TIMED OU`T

Cause:

Daemon could not connect to *`node`*.

Action:

Check the network.

`Error sending PID to master server`

Cause:

Vendor server could not send its PID to the top-level server in the hierarchy.

Action:

Report error to Prolifics technical support.

`f-do-notify called with no valid feature`

Cause:

Internal inconsistency error.

Action:

Report error to Prolifics technical support.

`Illegal connection request to` *`DAEMON`*

Cause:

A connection request was made to *`DAEMON`*, but this vendor daemon is not *`DAEMON`*.

Action:

Report error to Prolifics technical support.

`Illegal server connection request`

Cause:

A connection request came in from another server without a `DAEMON` name.

Action:

Report error to Prolifics technical support.

`KILL of child failed, errno =` *`nnn`*

Cause:

A daemon could not kill its child.

Action:

Get PID of daemon and kill with `kill -9`.

`No internet port number specified`

Cause:

Vendor daemon was started without an Internet port.

Action:

Specify an Internet port on the `SERVER` line.

`Not enough descriptors to re-create pipes`

Cause:

Top-level daemon detected the death of one of its sub-daemons. In trying to restart the chain of sub-daemons, it was unable to get the file descriptors to set up the pipes to communicate.

Action:

This is a fatal error. The daemons must be restarted.

`read: error message`

Cause:

An error in a `read` system call was detected.

`recycle_control BUT WE DIDN'T HAVE CONTROL`

Cause:

The hierarchy of vendor daemons has become confused over who holds the control token. This is an internal error.

`return_reserved: can't find feature listhead`

Cause:

When a daemon is returning a reservation to the free reservation list, it could not find the listhead of features.

`select: message`

Cause:

An error in a `select` system call was detected.

Action:

Report error to Prolifics technical support.

`Server exiting`

Cause:

Server is exiting; normally due to an error.

Action:

Report error to Prolifics technical support.

`SHELLO for wrong DAEMON`

Cause:

Vendor daemon was sent a `server` hello message that was destined for a different `DAEMON`.

```
Unsolicited msg from parent!
```

Cause:

Normally, the top-level vendor daemon sends no unsolicited messages. If one arrives, this message is logged. This is a bug.

Action:

Report error to Prolifics technical support.

```
WARNING: CORRUPTED options list (o->next == 0)
Options list TERMINATED at bad entry
```

Cause:

Internal inconsistency was detected in the daemon's option list.

Action:

Report error to Prolifics technical support.

APPENDIX

H Platform Notes

All Platforms

XML Libraries

Importing XML includes the use of libxml2, copyright 1998-2003 by Daniel Veillard.

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

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

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall Daniel Veillard be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

Notes for Motif

Tooltips

Tooltips in Motif use the LiteClue widget, copyright 1995 by Computer Generation, Inc.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall Computer Generation, Inc. nor the author be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

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

Notes for Microsoft Windows

The information in this appendix provides troubleshooting notes for installing and running Panther software on Microsoft Windows.

DLLs for JPEG Decoding

WECJLIB.DLL is a dynamic link library which performs JPEG decoding. There are two versions of the DLL. The version that comes with Panther can be freely distributed. An enhanced version, which has added functionality, can be ordered from Express Compression Labs (ECL).

In the version of the DLL provided with Panther, images are rendered using ordered dithering. The enhanced version of the DLL supports Hi-Color and Tru-Color display hardware. With such hardware, dithering is not necessary and the best possible picture quality can be achieved. Two-pass color quantization and improved dithering are also supported in the enhanced version, which results in better picture quality on 256-color display devices.

To order single- and multiple-user licenses for the enhanced version of the DLL, email: ecl@netcom.com, or write:

Dr. Y. Shan
P.O. Box 367
Caulfield East VIC 3145, Australia

Environment Space

If COMMAND.COM runs out of environment space when you issue a SET command, add the following line to your CONFIG.SYS file:

```
SHELL=C:\COMMAND.COM /E:1000 /P
```

If COMMAND.COM resides on a different drive or directory, modify the line accordingly.

Visual C++ Floating-Point Options

For Microsoft Visual C++ distributions, all the distributed libraries created in the current version of Panther are compiled with the /FPc switch, so that you can choose at link time which floating-point library to use. You can use either the math coprocessor library (LLIBC7.LIB), the emulator library (LLIBCE.LIB, the default), or the alternate math library (LLIBCA.LIB).

Visual Studio

A single `makefile` is provided for creating Panther executables. It is not necessary to use Microsoft Visual C++'s Visual Studio to create new Panther executables. Instead, you can invoke the `nmake` utility directly from the command line to create executables. If you want to use Visual Studio, you can use the `makefile` as an external module.

Using Panther Utilities

Because of the way Windows works, it can be challenging to use Panther's utilities when launched from the Program Manager or the Start menu. The problem arises because Windows' notion of the current directory is hidden from the user.

Most Panther's utilities are designed to be run from the command line. Under Windows, however, the current directory is usually set to the directory holding the executable, for example, `c:\prolifics\util`.

Therefore, if you launch a utility, such as `f2asc`, from the Program Manager or the Start menu and type `-a foo.asc foo.pro` in the Parameters window, `f2asc` looks for `foo.pro` in the `util` directory and creates its output `foo.asc` there as well. Since this is usually not what you will want, consider one of the following approaches:

- Enter full paths in the parameters window, for example, `-a d:\myprojoo.asc d:\myprojoo.pro`.
- Edit the properties on the Start menu to set the Working directory to where your files reside.
- Run the utilities from the command line or from a `makefile`.

Index

H

I

L

M

O

P

R

S

T

U

W