SYBASE®

Installation Guide

EAServer

6.1

[Sun Solaris]

Contents

About This Book

This book describes how to install EAServer, an application server, which includes an integrated set of development tools that you use to deploy Web applications that support high-volume traffic, dynamic content, and intensive online transaction processing (OLTP).

Audience   This book is for anyone responsible for installing EAServer.

How to use this book   Chapter 1, "Before You Install EAServer," contains system requirements and information about EAServer features and options.

Chapter 2, "Installing EAServer," contains instructions for installing and starting EAServer and Adaptive Server® Anywhere (ASA), upgrading an existing installation, and installing other EAServer products and features.

Chapter 3, "Creating Embedded EAServer Installations," explains how to install EAServer silently and use EAServer automated configuration tools to further configure the installation within your own installation script or program.

Chapter 4, "Installing and Configuring a Web Server Redirector Plug-In," describes how to set up a Web server redirector plug-in, which enables communication between the Web server and the EAServer HTTP and HTTPS protocol listeners.

Chapter 5, "Configuring Open Client and ODBC for EAServer," describes how to configure Open Client and ODBC drivers used with EAServer.

Related documents   **Core EAServer documentation**   The core EAServer documents are available in HTML and PDF format in your EAServer software installation and on the SyBooks™ CD.

EAServer 6.1 New Features Guide summarizes new functionality in this version.

The *EAServer API Reference Manual* contains reference pages for proprietary EAServer Java classes and C routines.

The *EAServer Automated Configuration Guide* explains how to use Ant-based configuration scripts to:

- Define and configure entities, such as EJB modules, Web applications, data sources, and servers
- Perform administrative and deployment tasks

The *EAServer CORBA Components Guide* explains how to:

- Create, deploy, and configure CORBA and PowerBuilder® components and component-based applications
- Use the industry-standard CORBA and Java APIs supported by EAServer

The *EAServer Enterprise JavaBeans User's Guide* describes how to:

- Configure and deploy EJB modules
- Develop EJB clients, and create and configure EJB providers
- Create and configure applications clients
- Run the EJB tutorial

The *EAServer Feature Guide* explains application server concepts and architecture, such as supported component models, network protocols, server-managed transactions, and Web applications.

The *EAServer Java Message Service User's Guide* describes how to create Java Message Service (JMS) clients and components to send, publish, and receive JMS messages.

The *EAServer Migration Guide* contains information about migrating EAServer 5.*x* resources and entities to an EAServer 6.*x* installation.

The *EAServer Performance and Tuning Guide* describes how to tune your server and application settings for best performance.

The *EAServer Security Administration and Programming Guide* explains how to:

- Understand the EAServer security architecture
- Configure role-based security for components and Web applications
- Configure SSL certificate-based security for client connections
- Implement custom security services for authentication, authorization, and role membership evaluation
- Implement secure HTTP and IIOP client applications
- Deploy client applications that connect through Internet proxies and firewalls

The *EAServer System Administration Guide* explains how to:

- Start the preconfigured server and manage it with the Sybase™ Management Console

- Create, configure, and start new application servers

- Define database types and data sources

- Create clusters of application servers to host load-balanced and highly available components and Web applications

- Monitor servers and application components

- Automate administration and monitoring tasks with command line tools

The *EAServer Web Application Programming Guide* explains how to create, deploy, and configure Web applications, Java servlets, and JavaServer Pages.

The *EAServer Web Services Toolkit User's Guide* describes Web services support in EAServer, including:

- Support for standard Web services protocols such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Uniform Description, Discovery, and Integration (UDDI)

- Administration tools for deployment and creation of new Web services, WSDL document creation, UDDI registration, and SOAP management

EAServer Troubleshooting Guide describes procedures for troubleshooting problems that EAServer users may encounter. This document is available only on the SyBooks Online Web site at:

http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.dc10113_0600/html/eastg/title.htm

jConnect for JDBC documents   EAServer includes the jConnect™ 6.0.5 driver to allow JDBC access to Sybase database servers and gateways. The *jConnect 6.0.5 Programmer's Reference* is available on the SyBooks Online Web site at:

http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.dc39001_0605/html/prjdbc/title.htm

Sybase Software Asset Management User's Guide   EAServer includes the Sybase Software Asset Management license manager for managing and tracking your Sybase software license deployments. The *Sybase Software Asset Management User's Guide* is available on the Getting Started CD and in the EAServer 6.0 collection on the SyBooks Online Web site at:

http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.dc00530_0200/html/sysamug/title.htm

Conventions The formatting conventions used in this manual are:

Formatting example	To indicate
commands and methods	When used in descriptive text, this font indicates keywords such as: • Command names used in descriptive text • C++ and Java method or class names used in descriptive text • Java package names used in descriptive text • Property names in the raw format, as when using jagtool to configure applications rather than the Web Console
variable, *package*, or *component*	Italic font indicates: • Program variables, such as *myCounter* • Parts of input text that must be substituted, for example: *Server*.log • File names • Names of components, EAServer packages, and other entities that are registered in the EAServer naming service
File | Save	Menu names and menu items are displayed in plain text. The vertical bar shows you how to navigate menu selections. For example, File | Save indicates "select Save from the File menu."
`package 1`	Monospace font indicates: • Information that you enter in the Web Console, a command line, or as program text • Example program fragments • Example output fragments

Other sources of information Use the Sybase Getting Started CD, the SyBooks CD, and the Sybase Product Manuals Web site to learn more about your product:

•   The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.

•   The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

• The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

To access the Sybase Product Manuals Web site, go to Product Manuals at http://sybooks.sybase.com/nav/base.do.

Sybase certifications on the Web
Technical documentation at the Sybase Web site is updated frequently.

❖ **Finding the latest information on product certifications**

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.

2 Select Products from the navigation bar on the left.

3 Select a product name from the product list and click Go.

4 Select the Certification Report filter, specify a time frame, and click Go.

5 Click a Certification Report title to display the report.

❖ **Creating a personalized view of the Sybase Web site (including support pages)**

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.

2 Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ **Finding the latest information on EBFs and software maintenance**

1 Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.

2 Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.

3 Select a product.

4   Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the "Technical Support Contact" role to your MySybase profile.

5   Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Accessibility features   EAServer has been tested for compliance with U.S. government Section 508 Accessibility requirements. The online help for this product is also provided in Eclipse help formats, which you can navigate using a screen reader.

The Web console supports working without a mouse. For more information, see "Keyboard navigation" in Chapter 2, "Management Console Overview," in the *EAServer System Administration Guide*.

The Web Services Toolkit plug-in for Eclipse supports accessibility features for those that cannot use a mouse, are visually impaired, or have other special needs. For information about these features see the Eclipse help:

1   Start Eclipse.

2   Select Help | Help Contents.

3   Enter `Accessibility` in the Search dialog box.

4   Select Accessible User Interfaces or Accessibility Features for Eclipse.

Note You may need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool.

For additional information about how Sybase supports accessibility, see Sybase Accessibility at http://www.sybase.com/accessibility. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

If you need help   Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

Before You Install EAServer

This chapter contains information you need to know before you install or upgrade EAServer.

Topic	Page
EAServer contents	1
EAServer documentation	2
System requirements	2
Installation types	3

EAServer contents

Besides the EAServer 6.1 software, the installation includes:

- SQL Anywhere® database server runtime files

- jConnect 6.05

- PowerBuilder 10.5 virtual machine (PBVM) to host PowerBuilder components

- Web Services Toolkit, which provides support for industry-standard Web services protocols and APIs such as SOAP, WSDL, and UDDI

- Web service support that allows you to deploy J2EE Web applications and EJBs as Web services

- Plug-ins for popular Web servers, which allow you to integrate EAServer Web applications with Web sites hosted on popular Web servers

- Various tools, management utilities, Java extensions, third-party JAR files, and sample applications used to extend EAServer capabilities

EAServer documentation

See the release bulletin for late-breaking installation information.

Online documentation is installed with EAServer. To view the documentation after installation, start EAServer, then in the Management Console, navigate to *http://hostname:8000/*, where *hostname* is the name of the machine that hosts EAServer. Select User Documentation.

You can also access the documentation on the SyBooks Online Web site at http://infocenter.sybase.com/help/index.jsp. To print these manuals, download the PDF version from the Web site.

System requirements

You can install EAServer 6.1 as a new installation or as an upgrade to an EAServer 5.*x* or 6.*x* installation. You can then migrate existing EAServer 5.*x* entities to EAServer 6.1. See the *EAServer Migration Guide* for instructions.

EAServer is supported on Sun Solaris versions 2.9 and 2.10; both require Sun patches. The memory requirement for all installation types is 512MB. The disk space requirements provided are approximate; the actual disk space required varies depending on options selected, cluster size, hard disk size, and partitioning. The installation program verifies that you have enough disk space and displays an error message if you do not have enough space to install the selected products and features.

The installation program requires 170MB of temporary disk space. The disk space for the installed software depends on the install type:

Installation type	Minimum hard disk space
Full	953MB
Client Runtime Only	250MB
Typical	455MB
Custom	Depends on features selected

Patches to install

To run EAServer on Solaris 2.9, install operating system patch number 111722-04. To get the patch, navigate to the SunSolve Web site at http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access, enter the patch number, and click Find Patch.

To run EAServer with JDK 1.4.2, verify that you have the required Solaris patches. See Solaris OS Patches at http://java.sun.com/j2se/1.4.2/download.html.

JDK versions

EAServer requires (and installs) versions 1.4.2, 1.5.0, and 1.6.0 of the Java Development Kit (JDK) from Sun Microsystems. You can also configure the installation to use an existing version, as long as it is of the required patch level.

Table 1-1: JDK versions and required patch levels

JDK version	Patch level
1.4.2	15
1.5.0	12
1.6.0	02

To verify the version and patch level, change to the *bin* directory of the JDK installation and run:

```
java -version
```

The installer prompts you to specify whether to use an existing JDK and for the location of the JDK. If the installer detects an existing JDK, the location displays as the default in this prompt. You can specify a different location.

Installation types

When you install EAServer, features are installed by default depending on the type of installation you select. The types of installations include:

• Typical – installs the typical files required to develop EAServer applications.

- Custom – you define the features to install. By default, features that are installed in a typical installation are selected for installation in a custom installation, but you can add or remove features.

- Client Runtime – installs only the client runtime files and Web server plug-ins. Select this option to run Java or C++ client applications or to use SSL connectivity from PowerBuilder clients, but not install the entire product set.

  You are not required to install the client runtime if you are developing clients on a machine where the application server is installed. The server installation includes the core server files.

  See Chapter 4, "Installing and Configuring a Web Server Redirector Plug-In," for more information about the Web server plug-ins.

- Add Components – adds components to an existing 6.1 installation.

- Upgrade – upgrades EAServer from version 6.0.0, 6.0.1, or 6.0.2.

- Full – installs all features.

Table 1-2 describes EAServer features and lists the features that are installed for each type of installation. A full installation installs everything.

Table 1-2: EAServer features

Feature	Description	Installation type
Server	Includes the application server and files required to manage the server and develop clients and components.	Typical
Core server files	Installs files required by the server.	Typical
Runtime libraries	Installs the required runtime libraries and includes these options: • Standard – compatible with JDK 1.4 and JDK 1.5. This is the default for a typical installation. • Optimized – compatible with JDK 1.5 only.	Typical
Client Runtime	Installs the client runtime files. Includes these options: • Standard – compatible with JDK 1.4, JDK 1.5, and JDK 1.6. JDK 1.5 is the default. • Optimized – compatible with JDK 1.5 only.	Client Runtime and Typical
Servlet Runtime	Installs the servlet runtime files.	Typical
ODBC Manager	Installs the Merant DataDirect Connect ODBC Driver Manager. See Chapter 5, "Configuring Open Client and ODBC for EAServer" for more information.	

Feature	Description	Installation type
Apache and Sun Java System Web Server plug-ins	Installs Web server plug-ins to integrate EAServer Web applications with Web sites hosted on popular Web servers. See Chapter 4, "Installing and Configuring a Web Server Redirector Plug-In," for more information.	Client Runtime
Administration	Installs: • J2EE Web Administration Console – a Web-based console that allows you to configure servers, packages, and components. • Web File Server – a Web application that provides FTP services on remote servers. • Web Shell – allows you to issue server and cluster commands through the shell command line. • Web Load Balancer – a Web application that balances server loads by redirecting HTTP requests from Web applications to application servers within a cluster.	Typical
Documentation	Installs online HTML documentation and tutorial files in the EAServer *html* subdirectory.	Typical
Extras	Installs these options: • jConnect 6.05 – JDBC (Java Database Connectivity) allows Java applications to access multiple database management systems using Structured Query Language (SQL). • JUDDI server – installs the files required to run a private Java implementation of the Uniform Description, Discovery, and Integration (UDDI) server. • Eclipse Development tool – installs an Eclipse plug-in to create and administer Web services, and optionally installs Eclipse.	Typical
JDKs	Options for installing JDK 1.4, 1.5, and 1.6.	JDK 1.4 and 1.5 are included in a typical installation
Open Client	Installs the Open Client™ software. See Chapter 5, "Configuring Open Client and ODBC for EAServer" for more information.	Typical
Optional Charsets	Allows you to install additional character sets.	
Samples	Installs the EAServer samples.	Typical

Feature	Description	Installation type
Sybase Asset Management System (SySAM)	(Optional.) Installs SySAM to perform license administration and asset management tasks. If you select this option, you can then select: • Network License Server – installs the network license server used to manage resource licenses. • License Utilities – installs SySAM license utilities. See the *Sybase Software Asset Management User's Guide*, available on the SyBooks Online Web site at http://infocenter.sybase.com/help/index.jsp.	Typical
Tools Support	Installs: • PowerBuilder 10.5.0 – support for running PowerBuilder 10.5.0 components in EAServer. • PowerBuilder 11.0 – support for running PowerBuilder 11.0 components in EAServer.	

CHAPTER 2   **Installing EAServer**

This chapter explains how to upgrade, install, and uninstall EAServer.

Topic	Page
Pre-installation tasks	7
Installing EAServer	8
Reinstalling or adding components	12
Upgrading your license	14
Installing SQL Anywhere	15
Starting EAServer, the Management Console, and SQL Anywhere	15
Uninstalling EAServer	16

Pre-installation tasks

Before you install EAServer:

1   If you plan to migrate existing EAServer 5.*x* entities to EAServer 6.1, see the *EAServer Migration Guide* for more information.

2   Read the release bulletins for the latest information on the products that you are installing.

    Release bulletins are on the Getting Started CD that is included with your software.

3   If you are reinstalling EAServer 6.1 or adding components or features, review the instructions described in "Reinstalling or adding components" on page 12.

4   If you are upgrading your license, see "Upgrading your license" on page 14.

5   Verify that you have enough space to install the software—see "System requirements" on page 2.

6   The installer requires a valid X-Windows configuration. Make sure the xterm command can run at the command prompt where you start the installer.

7   Make sure you have a TEMP environment variable and a *temp* directory with at least 170MB. The installation program uses the extra space and directory to write files temporarily during the installation. The installation program frees this space after the installation is complete.

8   Close any EAServer applications and application servers that run as Windows services.

Obtaining a license for EAServer   EAServer 6.1 uses runtime license management. This requires you to go to the Sybase Product Download Center at http://downloads.sybase.com at the end of the installation and get a license for any edition other than the Developer's Edition. A license is no longer supplied with your software.

Installing EAServer

Installing EAServer takes 10 – 20 minutes, depending on the speed of your computer and features you are installing.

What the installer does   The setup program installs software in default drives and directories unless you specify otherwise. If a directory you specify does not exist, the setup program creates it.

❖   **Installing EAServer**

1   Exit any programs that are running. If you do not, the Sybase installer may not be able to copy some files to the appropriate directories.

2   If you have downloaded EAServer, expand the installation software to a temporary location. Otherwise, insert the software disk into your CD drive.

3   To start the installer from the command line, change to the location of the installation software and enter:

```
./setup [-is:tempdir temp_directory]
```

If you have less than 170MB in your temp directory, add the -is:tempdir *temp_directory* option, replacing *temp_directory* with the full path to a temporary directory that has at least 170MB of available disk space.

The installer starts, and the EAServer 6.1 Install window appears.

4   Click Next.

Note Use Back and Next to step backward and forward through the installation process to modify specifications as necessary.

Select Cancel to halt the installation process.

---

5   Select your country or region from the drop-down list to display the license agreement. You must read and accept the terms of the license agreement for your country before you can install any Sybase products. Click Next.

6   The installer checks whether any EAServer processes are running on the machine. If prompted, shut down any EAServer applications, including EAServer. Click Next.

7   Enter the full path to which EAServer is to be installed.

To modify the default locations of either the Sybase or Shared directory locations, select Show Advanced Locations.

8   Select the type of installation, then click Next:

- • Typical – installs EAServer using the most common installation options.

- • Client Runtime – installs the files required to run standalone EAServer clients.

- • Full – installs everything.

- • Custom – allows you to choose specific installation options for EAServer.

- • Add Components – adds components to an existing 6.1 installation.

- • Upgrade – upgrades EAServer from version 6.0.0, 6.0.1, or 6.0.2.

See "Installation types" on page 3 for a list of features that are installed for each installation type.

9   Select a JDK check box to install the JDK.

To use an existing JDK, enter a path (or use the Browse button) to locate an existing JDK that may already be installed on your system. If the installer detects an existing JDK of the appropriate version, it displays as the default location. Existing JDKs must be of the correct version and patch level, as described in "JDK versions" on page 3.

Click Next to continue.

10 If you are creating a new installation, the installer displays port numbers for the preconfigured EAServer network listeners, as displayed in Table 2-1. You can edit the ports or accept the default settings. Choose port numbers that are not used by other servers running on your machine.

You can click "Suggest ports" to test whether any of the specified ports are in use. The installer tests each port to see if it is used by a running process or in known EAServer installations on the machine. If a port is in use, the number is incremented by 100 and tried again.

Click Default to restore the ports to their default values.

When you click Next, the installer also tests whether ports are in use. After the installation completes, you can modify or remove preconfigured listeners as described in Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

Table 2-1: Preconfigured listeners

Listener name	Default port	Description
http	8000	For HTTP (Web client connections).
https1	8001	HTTP with SSL security.
https2	8002	HTTP with SSL security requiring client certificates to connect.
iiop	2000	For IIOP connections. The Management Console, PowerBuilder, and other standalone client applications use this protocol.
iiops1	2001	IIOP with SSL security.
iiops2	2002	IIOP with SSL security requiring client certificates to connect.
Hostname	*host*	The host on which you are installing EAServer.
rmi	1999	Java Remote Method Invocations (RMI).

11 The installer displays a summary of the features to be installed and the installation directory. Review these entries and click Install to continue or Back to modify your entries.

12   The installer begins copying files.

13   Select the product edition and license type from the corresponding drop-down lists. If you do not know this information, ask your system administrator.

   •   **License product edition**

Product edition	Description
AE	Advanced Edition
DE	Developer Edition
WE	Workgroup Edition

   •   **License type**

License type	Description
AC	Application Deployment CPU.
AO	Application Deployment Other.
BC	Application Deployment Standby CPU.
CP	CPU-based license. Allows users to install and run the software on a machine with a specified number of CPUs. No restrictions exist on either the number of software copies that users can run on the specified machine, or the number of users that can connect to the software.
DT	Development and Testing.
OT	Other.
SF	Standby CPU.
SR	Server-based license. Enables users to install and run the software on the specified server. In addition to the server license, users must also purchase a Concurrent-User or Network-Seat license.
SS	Standalone Seat.

14   The installer prompts you for an administrative password. Enter and confirm a password for the admin@system user following the guidelines described by the installer. *Remember this password.* When you log in to the Web Console and connect to EAServer, you will need to provide the user name and password.

15   A summary screen informs you when the installation is complete. If you do not yet have a license, you must obtain one by selecting the Launch the Sybase Product Download Center (SPDC) Web Site option.

   Select any of these options to perform the corresponding tasks at this time. You can also do any of these tasks later:

- Start Server – starts EAServer.

- View Readme – displays the readme file.

- Launch the Migration Tool – allows you to migrate EAServer 5.*x* entities to EAServer 6.1. See the *Migration Guide* for details.

- Launch the Sybase Product Download Center (SPDC) Web Site – log in to the SPDC Web site to obtain a license for EAServer. You must know the product edition and license type for your particular installation. If you do not have this information, ask your system administrator. See the *FLEXnet Licensing End User Guide*, and the *Sybase Software Asset Management User's Guide* for more information.

16 Click Finish to exit the installer.

For each of the following products you choose to install, see the corresponding section for additional installation and configuration instructions:

- SQL Anywhere support – "Installing SQL Anywhere" on page 15.

- Any of the Web server redirector plug-ins (Apache or Sun Java System) – Chapter 4, "Installing and Configuring a Web Server Redirector Plug-In."

- Open Client – Chapter 5, "Configuring Open Client and ODBC for EAServer."

- jConnect – the jConnect user documentation.

Network install   If you are a system administrator, you may want to allow users to install EAServer from a network drive rather than from the CD. To do this, copy the files from each CD to a separate directory on a network drive and note the location. During installation, the user must specify the directory location for each CD.

Reinstalling or adding components

This section describes how to add or reinstall components to a current version installation.

To add components to an existing installation, or to reinstall EAServer, run the *setup* program as described in "Installing EAServer" on page 8, and select an installation directory that contains an EAServer installation.

❖ **Adding components**

1 The installer prompts you for the type of installation: Client Runtime or Custom. Select Client Runtime if you are adding components to a client installation. Select Custom for all other types of installations. Click Next.

2 Select the components you want to add.

3 Verify that the EAServer directory is correct and that the components being added are correct. Click Next.

4 You may be prompted to overwrite existing files. Click Yes to continue.

5 You see these:

• License server – if you are not obtaining licenses from a license server, select No and Next. If you select Yes, enter the name of the host name and port number of the license server from which you obtain your license. See the *FLEXnet Licensing End User Guide*, and the *Sybase Software Asset Management User's Guide* for more information.

• Configure the SySAM e-mail alert mechanism. Select No and Next, or Yes to configure the SYSAM e-mail alert mechanism. See the *Sybase Software Asset Management User's Guide*, for more information.

• Select the product edition and license type from the corresponding drop-down lists. If you do not know this information, ask your system administrator.

• The installer prompts you for an administrative password. Enter and confirm a password for the admin@system user following the guidelines described by the installer. *Remember this password*. When you log in to the Web Console and connect to EAServer, you will need to provide the user name and password.

6 When the components have been added, the installer displays a message that EAServer has been successfully installed. Click Finish.

❖ **Reinstalling EAServer**

1 The installer prompts you for the type of installation: Typical, Client Runtime, Full, or Custom. If you select Typical or Full, a default set of features is reinstalled. Custom and Client Runtime allow you to modify your selections. Click Next.

2 Select the components you want to reinstall.

3 You see these prompts:

- License server – if you are not obtaining licenses from a license server, select No and Next. If you select Yes, enter the name of the host name and port number of the license server from which you obtain your license. See the *FLEXnet Licensing End User Guide*, and the *Sybase Software Asset Management User's Guide* for more information.

- Configure the SySAM e-mail alert mechanism. Select No and Next, or Yes to configure the SYSAM e-mail alert mechanism. See the *Sybase Software Asset Management User's Guide* for more information.

- Select the product edition and license type from the corresponding drop-down lists. If you do not know this information, ask your system administrator.

- The installer prompts you for an administrative password. Enter and confirm a password for the admin@system user following the guidelines described by the installer. *Remember this password*. When you log in to the Web Console and connect to EAServer, you must provide the user name and password.

4 The installer displays a message when EAServer has been successfully reinstalled. Select Start Server to start EAServer or View Readme to display the readme file. Click Next.

5 Click Finish to exit the installer.

Upgrading your license

To upgrade your license, go to the Sybase Product Download Center at http://downloads.sybase.com. You must know the product edition and license type for your particular installation. If you do not have this information, ask your system administrator. You do not need a license for the Developer Edition of EAServer. All other editions require a license.

See the *FLEXnet Licensing End User Guide*, and the *Sybase Software Asset Management User's Guide* for information about upgrading your license, changing license servers, and so on.

Installing SQL Anywhere

SQL Anywhere is installed as part of a typical EAServer installation. It is installed in the *ASA100* subdirectory under the EAServer installation directory.

Starting EAServer, the Management Console, and SQL Anywhere

This section describes how to start EAServer, the Management Console, and the SQL Anywhere database.

❖ **Starting EAServer**

• Change to the EAServer *bin* directory, and run start-server; for example:

```
cd /work/Sybase/EAServer/bin
start-server.sh
```

The server starts and runs as a foreground process in the current console window. To start the server in a separate window, use the -xterm option; for example:

```
cd /work/Sybase/EAServer/bin
start-server.sh -xterm
```

The server starts in a new X terminal window. To use this option, you must have installed and configured X-Windows. In other words, the xterm command must successfully launch a terminal window in the shell where you start the server.

There are other ways to start the server. For details, see "Starting the server" in Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

With EAServer running, you can start the Management Console to configure new servers, packages, components, and so on.

❖ **Starting the Management Console**

The Management Console allows you to configure servers, packages, and components. You can run the Management Console in either a Web browser or Eclipse. With EAServer running:

1 To run the Management Console in a Web brower, launch the Web browser and enter the URL of the Management Console:

```
http://hostname:port/console
```

where *hostname* is the host on which EAServer resides, and *port* is the HTTP port used to receive client connections.

To run the Management Console in Eclipse: start Eclipse, select Help | Help Contents, then choose the Management Console.

2   Enter `admin` as the user name, and the password that you established for this user during the installation, then click Login. You can reset this password using the set-admin-password command, which is located in the *bin* subdirectory of your EAServer installation.

3   Select any of the installed plug-ins to access EAServer resources.

Note When you install EAServer, the default host name for all listeners is set to the name of the host machine and the default IIOP port number to 2000. After connecting to EAServer, you can use the Management Console to edit the listener properties and change the default host name and port numbers. Some of the sample programs use "localhost" as the host name when connecting to EAServer. See "Configuring listeners" in Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide* for more information.

❖ **Starting SQL Anywhere**

After you have started EAServer, you must start SQL Anywhere before you can run any sample application.

1   Make sure the xterm utility runs in your environment.

2   To start the SQL Anywhere database server, go to the *bin* subdirectory of your EAServer installation and run:

```
./asa-start.sh
```

Uninstalling EAServer

When you install EAServer, the installer creates a *_uninstall* subdirectory that contains the *EASuninstall_sol* executable used to uninstall EAServer.

❖ **Running EASuninstall_sol**

1   Verify that EAServer is shut down before you run *EASuninstall_sol*.

2   If you plan to reinstall EAServer, make backup copies of files you may want to reuse, such as the DB, INI, and PBL files.

3   From the *_uninstall* subdirectory, enter:

```
./EASuninstall_sol
```

This launches the Sybase uninstaller. Click Next in the Sybase uninstaller wizard.

4   Select Uninstall in the summary panel to uninstall EAServer.

5   The uninstaller displays a message when it has uninstalled EAServer. Click Finish.

Note The uninstall program does not remove any server or user created files from an installation upgrade, such as log files, repository files, stubs, skeletons, and temporary work files, or backup files created during the installation process. You can manually remove these files after running the uninstaller.

Creating Embedded EAServer Installations

If you are packaging EAServer with your own software, you may want to use a script to create a silent installation, so your end users install EAServer with no interaction with the EAServer installer.

Additionally, EAServer provides tools for scripted configuration of the installation, including customization of server properties, and installation of your own components and application files to EAServer.

Configuring and running the silent installer

The silent installer is a Java program. The installation CD contains a sample batch file to run the install with the correct JRE and CLASSPATH settings. An additional text file specifies the installation type and options. These instructions assume that you will include the EAServer install files and customized installer scripts with the install media for your own software.

❖ **Configuring the silent installer**

1   Create a directory for your install image.

2   Copy the following files from the EAServer installation CD to the install image directory:

  •   *EAS610.jar*.

  •   *readme.htm*. If you are installing the HTML documentation, this file contains additional documentation.

- The *JRE_1_5* subdirectory and its contents. Sybase recommends that you use this JRE to run the install. Other JRE versions may not work as well.

- The *Modules/LicensePanel* directory is required, as are the other files and directories in the Modules folder.

3   Create configuration and run files as described in "Configuration and run files" on page 20. Copy any required additional files from the EAServer CD to your install image, as described in that section.

4   To support the silent uninstallation process, create the files described in "Silent uninstaller files" on page 28.

❖   **Testing and running the silent installer**

1   Before running a silent installation, make sure you:

a   Have enough space in your temporary and product directories. See "System requirements" on page 2. If necessary, you can edit the shell script that runs the install and change the temporary directory location specified for the -Dtemp.dir command line option.

b   If your home directory contains an InstallShield *vpd.properties* file, make a backup copy. If you run the install with a different user ID, check for this file in the home directory of that user ID and back it up if it exists.

2   Test the silent installer using the run script that you created at the command line or in your own product's installation script. Running a silent installation takes 10 – 20 minutes, depending on the speed of your computer.

3   After each trial run, check for errors, and clean up your machine as described in "Troubleshooting and cleanup of the silent installer" on page 30.

Configuration and run files

In a silent installation, users cannot input information or choices. You must supply all required information in a configuration file or on the command line that runs the silent install. The EAServer installation script contains a sample configuration file, *SilentInstall.txt*. The shell script *SilentInstall.sh* runs the install with this configuration. Start with copies of these files and modify them to suit your installation.

Place your configuration file and run script in the root directory of your install image. Edit the run script to refer to the file name you are using for your configuration file. You may need to change the temporary directory location specified for the -Dtemp.dir command line option. This directory must have enough space for the installer's temporary work files, as described in "System requirements" on page 2.

Edit the configuration file to customize the install as described below.

Product editions and license types

You can define various product installation and license types by defining the Sybase software asset management product edition and license type. Table 3-1 lists allowable values.

Table 3-1: product editions and license types

Variable	Value

-V Variable_CBEASSySAMEdition	Specifies the product edition:
	`"Advanced Edition"`
	`"Workgroup Edition"`
	`"Developer Edition"`

-V Variable_CBEASSySAMType	If you are using a license server, include the Sybase Software Asset Management product edition and license type. To modify the product edition and license type, use these values when setting
	-V Variable_CBEASSySAMedition=:
	`"Application Deployment CPU License (AC)"`
	`"Application Deployment Other License (AO)"`
	`"Application Deployment Standby CPU License (BC)"`
	`"CPU License (CP)"`
	`"Development and Testing License (DT)"`
	`"Other License (OT)"`
	`"Standby CPU License (SF)"`
	`"Server License (SR)"`
	`"Standalone Seat License (SS)"`

Installation location

Set the `-W setInstallLocUnix.value` to specify the installation location.

The default value is `$D(install)/sybase`, which specifies the directory */opt/sybase*.

License agreement

For the silent install to run, you must change the value of -V AgreeToSybaseLicense from false to true, indicating that you have read and agreed to the software license agreement. You can view license text by running the interactive install or on the Sybase Web site at http://www.sybase.com/softwarelicenses.

Listener ports

EAServer includes several preconfigured network listeners for the supported protocols. For new installations, configure the listener ports by setting the parameters listed in Table 3-2. You can also configure these settings after the installation completes using the Management Console, described in the *EAServer System Administration Guide*, or Ant or jagtool, described in the *EAServer Automated Configuration Guide*.

Table 3-2: Silent installer listener port parameters

Parameter	Specifies
-V Variable_TFPortiiop	The initial port for the IIOP listener
-V Variable_TFPortiiops1	The initial port for the IIOPS1 listener
-V Variable_TFPortiiops2	The initial port for the IIOPS2 listener
-V Variable_TFPorthttp	The initial port for the HTTP listener
-V Variable_TFPorthttps1	The initial port for the HTTPS1 listener
-V Variable_TFPorthttps2	The initial port for the HTTPS2 listener
-V Variable_TFPortrmi	The initial port for the RMI listener
-V Variable_TFPorthostname	The host name to which the port numbers are assigned

JDK installation parameters

You can configure EAServer to use the JDK versions described in "JDK versions" on page 3. For each JDK version, you can install the JDK, use an existing installation, or not use that JDK version at all.

The parameters in Table 3-3 allow you to configure the EAServer installation to use JDK installations that are already in place, rather than installing JDKs with the EAServer installation.

Table 3-3: Silent installer existing JDK parameters

Parameter	Specifies
-V EASJDKUseExisting_JDK14_CheckBox	Whether to use an existing JDK 1.4 installation. To use an existing installation, set this parameter to true and specify the location as the value of the next parameter. Also, set the value of JDK14.active to false in the feature selection section.
-V EAS_JDK14_Install_Location	If you are using an existing JDK 1.4 installation, the location where it is installed. Verify the version and patch level of the specified JDK as described in "JDK versions" on page 3.
-V EASJDKUseExisting_JDK15_CheckBox	Whether to use an existing JDK 1.5 installation. To use an existing installation, set this parameter to true and specify the location as the value of the next parameter. Also, set the value of JDK15.active to false in the feature selection section.
-V EAS_JDK15_Install_Location	If you are using an existing JDK 1.5 installation, the location where it is installed. Verify the version and patch level of the specified JDK as described in "JDK versions" on page 3.
-V EASJDKUseExisting_JDK16_CheckBox	Whether to use an existing JDK 1.6 installation. To use an existing installation, set this parameter to true and specify the location as the value of the next parameter. Also, set the value of JDK15.active to false in the feature selection section.
-V EAS_JDK16_Install_Location	If you are using an existing JDK 1.6 installation, the location where it is installed. Verify the version and patch level of the specified JDK as described in "JDK versions" on page 3.

To install a JDK from your install image, enable the feature parameter for that JDK and include the required files in your image, as listed in Table 3-4.

Table 3-4: Parameters to install JDKs

Parameter	Specifies	Comments
-P JDKs.active	Whether to install any JDKs from the install image	Must be set to true or the next two parameters are ignored
-P JDK14.active	Whether to install JDK 1.4	
-P JDK15.active	Whether to install JDK 1.5	
-P JDK16.active	Whether to install JDK 1.6	

Sybase Software Asset Management License input parameters

If licenses are to be obtained from the License Server, you must define the license server parameters. Set the parameters in Table 3-5 to define the license server.

Table 3-5: License server parameters

Parameter	Set the value to
-V Variable_LicServerYes	Set to true to use a license server
-V Variable_LicServerHostname	The license server host
-V Varaiable_LicServerPortNum	The license server port

Sybase Software Asset Management e-mail alerts

To configure e-mail alerts, set:

-V Variable_RBEmailAlertsYes=true
-V Variable_RBEmailAlertsNo=false

Uncomment and set the -V Variable_CBSySAMEmailSeverity variable to one of these values:

WARNING
INFORMATIONAL
ERROR

Additional SySAM variables that you can set include:

• -V Variable_TFSySAMEmailHost=smtp

• -V Variable_TFSySAMEmailPort=25

• -V Variable_TFSySAMEmailSender=" "

• -V Variable_TFSySAMEmailRecipient=" "

• -V Variable_CBSySAMEmailSeverity=

See the *Sybase Software Asset Management User's Guide* for additional information.

Feature selection parameters

These parameters specify which optional features to install. Table 3-6 lists the parameters that select which EAServer features are installed. Each parameter requires a value. Specify true to install the feature or false to not install the feature.

Some features have a parent-child relationship (shown by indentation in the sample installation script). To install child features, you must enable both the parent feature and the child feature.

Some features require additional files to be added to your installation image, as listed in Table 3-6. If you enable these features, add the required files to your image by copying them from the EAServer installation CD. Paths within your install image must match those listed in Table 3-6.

Table 3-6: Feature selection parameters

Parameter	Feature	Additional requirements
-P Server.active	Parent feature for several core server and client install features.	
-P CoreServerFiles.active	Files required to run servers.	Requires parent feature -P Server.active. Requires files: */Modules/EAS5_Patches/5xMigrationPatches.jar* *Modules/ASA100.jar* *Modules/EAS-60_ThirdPartyLegal.pdf*
-P RuntimeLibraries.active	Parent feature for client runtime libraries. No runtime libraries are installed unless this parameter is set to true.	Requires parent feature -P Server.active.
-P Standard.active	Standard is compatible with JDK1.4 and JDK1.5.	
-P Optimized.active	Optimized is compatible with JDK1.5 only.	
-P ClientRuntime.active	Installs the client runtime files.	Requires parent feature -P RuntimeLibraries.active. Requires file: */Modules/EAS-60_ThirdPartyLegal.pdf*
-P Standard_Client.active	Standard client is compatible with JDK1.4 and JDK1.5.	Requires parent feature -P ClientRuntime.active.
-P Optimized_Client.active	Optimized client is compatible with JDK1.5 only.	Requires parent feature -P ClientRuntime.active.
-P ServletRuntime.active	Installs the servlet runtime files.	Requires parent feature -P RuntimeLibraries.active. Requires file: */Modules/servlet-rt.jar*
-P ODBCManager.active	Installs the ODBC manager.	

Parameter	Feature	Additional requirements
-P WebServerPlugins.active	Parent feature for Web server plug-in files. No Web server plug-ins are installed unless this parameter is set to true.	
-P EASApache20.active	The Apache 2.0 Web server redirector plug-in.	Requires parent feature -P WebServerPlugins.active. Requires files: */Modules/redirector-apache-2.0.jar* */Modules/redirector-common-files.jar*
-P SunJavaSysWebServ61.active	The Sun Java System 6.1 Web server redirector plug-in.	Requires parent feature -P WebServerPlugins.active. Requires files: */Modules/redirector-sun-java-web-server-6.1.jar* */Modules/redirector-common-files.jar*
-P Administration.active	Parent feature for administration features.	
-P J2EEWebAdminCons.active	Installs files required to run the Web Administration console.	Requires parent feature -P Administration.active. Requires files: */Modules/Help-Eclipse.jar* */Modules/RT14/application-console.zip*
-P WebFileServer.active	Installs files required to run the Web file server.	Requires parent feature -P Administration.active. Requires file: */Modules/RT14/wfs.war*
-P WebShell.active	Installs files required to run the Web shell.	Requires parent feature -P Administration.active. Requires file: */Modules/RT14/wsh.war*
-P WebLoadBalancer.active	Installs files required to run the Web load balancer.	Requires parent feature -P Administration.active. Requires file: */Modules/RT14/wlb.war*
-P Documentation.active	Installs EAServer documentation in HTML format in the server's default document root location.	Requires files: */Modules/docs-generic.jar*
-P Extras.active	Parent feature for extra features.	

Parameter	Feature	Additional requirements
-P jConnect605.active	Installs the jConnect DB Scripts.	Requires parent feature -P Extras.active.
-P JUDDIServer.active	Installs the JUDDI server.	Requires parent feature -P Extras.active. Requires file: */Modules/Extras-Juddi.jar*
-P EclipseDevToolPlugin.active	Installs the Eclipse development tool plug-in.	Requires parent feature -P Extras.active. Requires files: */Modules/eclipse-SDK-3.1.1-solaris-motif.zip* */Modules/syeclipse.jar* */Modules/syeclipse-help.jar* */Modules/RT14/sybasewst.war*
-P JDKs.active	Parent feature for JDK installation.	
-P JDK14.active	Installs JDK 1.4.	Requires parent feature -P JDKs.active. Requires file: */Modules/eas-jdk-14.jar*
-P JDK15.active	Installs JDK 1.5.	Requires parent feature -P JDKs.active. Requires file: */Modules/eas-jdk-15.jar*
-P JDK16.active	Installs JDK 1.6.	Requires parent feature -P JDKs.active. Requires file: */Modules/eas-jdk-16.jar*
-P OpenClient.active	Installs Sybase Open Client libraries and support files (required to use ODBC connection caches that connect to Sybase databases if you do not have Open Client installed already).	Requires file: */Modules/opentclient.jar*.
-P OptionalCharsets.active	Installs additional character sets.	Requires file: */Modules/charsets.jar*
-P Samples.active	Installs sample programs.	Requires files: */Modules/samples-generic.jar* */Modules/samples-sol.jar*
-P SybaseSYSAM.active	Parent feature for Sybase Software Asset Management (SySAM)	

Parameter	Feature	Additional requirements
-P NetworkLicenseServer.active	Installs the SySAM network license server.	Requires parent feature -P SybaseSYSAM.active Requires file: */Modules/sysam.jar*
-P LicenseUtils.active	Installs the SySAM license utilities.	Requires parent feature -P SybaseSYSAM.active Requires file: */Modules/sysam-utils.jar*
-P ToolsSupport.active	Parent option for the tools support option. If this parameter is set to false, the tools support options are ignored.	
-P PowerBuilderv1051.active	Installs the PowerBuilder version 10.5.1 virtual machine.	Requires parent feature -P ToolsSupport.active. Requires file: */Modules/pbvm1051.jar*
-P PowerBuilderv11.0.0.active	Installs the PowerBuilder version 11.0 virtual machine.	Requires parent feature -P ToolsSupport.active. Requires file: */Modules/pbvm1100.jar*

Specifying parameters on the command line

You may want to configure some install settings dynamically at install time. For example, you may want to set the EAServer install directory to a location selected by the end user of your own installer. To do this, you can remove settings from the configuration file and specify them as command line arguments to the silent install.

For example, if your silent install script is *SilentInstall.sh*, this command installs EAServer to */opt/Sybase/EAServer*:

```
SilentInstall.sh -W "setInstallLocUnix.value=/opt/Sybase/EAServer"
```

You must also remove the equivalent settings from the silent install configuration file.

Silent uninstaller files

You can configure the silent installer to support silent uninstallation. This creates a script that your users can run to silently remove the installation from their system. The silent uninstaller requires:

- The *EASuninstall.jar* and *EASuninstall.dat* files that are created when users run the install. The JAR file contains the Java uninstallation program, and the *.dat* file contains data about installed features. These files are installed in the *_uninstall* subdirectory of your EAServer installation.

- A JRE installation of the same version as found on the EAServer install CD.

- The files *SilentUninstall_Unix.txt* and *SilentUninstall_Unix.sh*. If these are present in the root directory of your silent installer, they are copied to the *_uninstall* directory when users run the install. You must prepare these files as described below before you release your silent installer to your users.

SilentUninstall_Unix.txt

A sample of this file is on the EAServer installation CD. However, *SilentUninstall_Unix.txt* does not run unless you edit the copy placed in your install image. This file configures the features to uninstall, using syntax similar to the options described in "Feature selection parameters" on page 24. To remove everything, set all the feature options to true. For a partial uninstallation, change the feature options to false for those features that should not be removed.

Note Some feature options in *SilentUninstall_Unix.txt* have a parent-child relationship, indicated by indentation in the sample file. To uninstall a parent feature, the parent feature and all child features must be set to true in *SilentUninstall_Unix.txt*.

SilentUninstall_Unix.sh

Users will run this file to remove the installation from their systems. A sample of this file is on the EAServer installation CD. Place a copy in your install image, and verify the following, keeping in mind that the file will be run in the EAServer *_uninstall* directory:

- The CLASSPATH includes *EASuninstall.jar* (located in the same directory).

- The java command line specifies the path to a java executable of the same version as supplied on the EAServer software CD. You can run the uninstall with the JRE that is installed in the *_jvm* subdirectory of the installation, for example:

```
../_jvm/bin/java -classpath $CLASSPATH -Dtemp.dir=/tmp run \
-options SilentUninstall_Unix.txt
```

You may need to change the directory specified for temporary work files, specified by the -Dtemp.dir command line option.

The uninstaller does not remove all files. Files created after the install was run are not deleted. This includes log files, property and resource files updated at runtime, and any application files that you have created in the EAServer directory. After uninstalling, you must remove these files manually.

Troubleshooting and cleanup of the silent installer

After a trial run of your silent install, check for errors, verify the installation, and clean up the machine before trying another run.

Check for installer errors on the console and in the installer log file in the specified install location. If you see *ZipException* errors, make sure you have included all required files in the install image.

When testing your installation results, start a server in the installation and verify that the expected features are licensed by checking the licensed features listed in the server log file. If not, verify that you have configured the license parameters described in "Sybase Software Asset Management License input parameters" on page 23.

Test any other features that you are installing, such as the Web Console, Web Services, and so forth.

❖ **Cleaning up the machine**

Before re-running a silent installation, uninstall the previous installation using the graphical or silent uninstaller. For graphical uninstaller instructions, see "Uninstalling EAServer" on page 16. For silent uninstaller instructions, see "Silent uninstaller files" on page 28.

If your installer was not configured correctly, or you abort the installation before it completes, the uninstall process may fail. In that case, clean the previous installation from your machine as follows:

1   Delete the EAServer installation directory and subdirectories.

2   If you made a backup copy of the *~/vpd.properties* file, restore it. Otherwise, delete the *~/vpd.properties* file that was generated during the install.

3   Make sure the JAGUAR or DJC_HOME environment variables are not set in the shell where you re-run the install.

Post-installation configuration

EAServer provides several options for scripted automated configuration of server properties and application components:

*   jagtool is a command line configuration program that you can call from install scripts or batch files.

*   jagant allows you to configure EAServer from Jakarta Ant scripts.

For information on jagtool and jagant, see "Using jagtool and jagant" in the *EAServer Automated Configuration Guide*.

You can call these tools from your installation script or install program to configure the server and install application components. For example, you might install your own components by importing them as EAServer JAR files with the jagtool or jagant deploy command, then use the install command to install the components to the preconfigured server. You might also use the create command to define data sources (connection caches).

Most tasks that you can accomplish manually from the Management Console can be automated using jagtool or jagant. In addition, the *EAServer Automated Configuration Guide* explains how to use Ant-based configuration scripts for configuring embedded EAServer installations.

You can create a single JAR file that contains a server configuration, data source configurations, components, Web applications, and all other entities required to run your application.

To create a deployable JAR file, use the jagtool export command. At install time, you can use the jagtool deploy command to install the entities into the EAServer repository.

You can install additional required files, such as the shell script that runs your server (in the EAServer *bin* directory), and JAR files that contain utility Java classes or JDBC drivers (in the EAServer *java/lib* directory). To add files to an entity collection, list them in the com.sybase.jaguar.entitycollection.files property. They are installed when you deploy the JAR file.

Installing and Configuring a Web Server Redirector Plug-In

The Web server redirector plug-in allows communication between the Web server and the EAServer HTTP and HTTPS protocol listeners.

Overview

The EAServer application server hosts Web applications, and functions as a Web server. The redirector plug-ins are installed on third-party Web servers, and redirect requests to EAServer (for example, Apache to EAServer). You can also use redirectors to perform load balancing and failover. The Web server forwards requests that need to access EAServer servlets, JSPs, and so on, to EAServer. EAServer processes the requests and returns the results back to the Web server.

You can establish HTTPS (HTTP over SSL) connections between your Web server redirector and EAServer. This provides an SSL communication link (data integrity and encryption services), but does not support authentication services using client-side certificates.

Table 4-1 describes the Web servers for which a redirector plug-in is available, and the EAServer files required to enable them.When setting up a redirector for a specific Web server, simply unzip the appropriate redirector JAR file and the *redirector-common-files.jar* into the appropriate location for that Web server.

Table 4-1: Redirector plug-in files

Web Server	File name	Description
Apache 2.0 version 55	*redirector-apache-2.0.jar*	Apache 2.0 specific redirector files
Sun Java System Web Server 6.1	*redirector-sun-java-web-server-6.1.jar*	Sun Java Web Server 6.1 specific redirector files
All	*redirector-common-files.jar*	Files that are common for all redirectors

Table 4-2 lists variables and expressions that are used throughout this chapter.

Table 4-2: Local variables and expressions

Term	Description
Application server	Refers to a single installation of EAServer
WEB_SERVER_HOME	Represents the installation directory of your Web server

How the plug-in works

As Figure 4-1 on page 35 illustrates, the redirector plug-in forwards HTTP requests to EAServer. For Web servers to forward requests, you must:

1 Install the redirector plug-in on the Web server.

2 Edit the Web server configuration file to:

- Locate and load the redirector plug-in.

- Configure context paths that define which requests (application server hosted Web applications, JSPs, servlets, and so on) are forwarded to EAServer.

3   Create and edit the redirector configuration file to define the application servers to which the redirector sends requests.

**Note** Quote absolute path names containing spaces. Use forward slashes in path names, not backslashes.

4   Configure EAServer so it accepts requests from the redirector.

Configuration differs by Web server   Although the process for installing and configuring the redirector plug-in is similar for each Web server, the names of the redirector plug-in file, Web server configuration file, and the redirector configuration file (which you create) vary. See the section for your particular Web server when installing and configuring the redirector plug-in.

Figure 4-1: HTTP request flow through the redirector
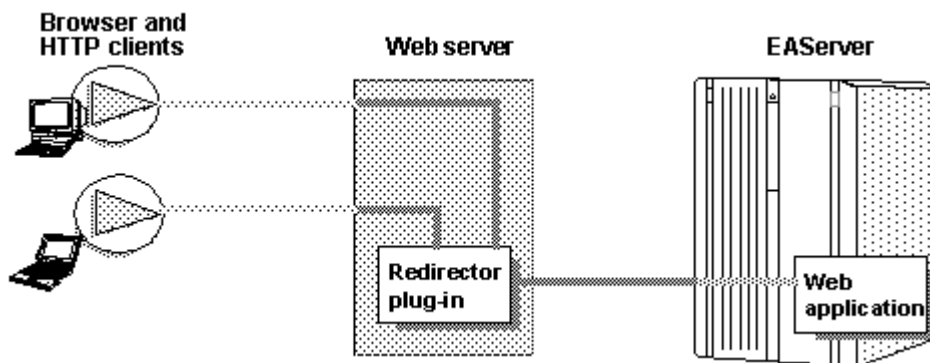
Load balancing

Load balancing is transparent to the client. The redirector detects and directs requests from existing sessions to EAServer using either the J2EE cookie or the J2EE session identifier that is part of the HTTP request.

If the cookie or identifier is missing, the client is treated as a new client, otherwise, the redirector sends the request to the EAServer that originally established the session.

New requests are directed to different application servers that share and balance the load. Load balancing is based on a round-robin scheme. If a redirector is configured to service *n* number of application servers, and a new request arrives at the redirector, the request is directed to the next application server in its list (as defined by the redirector configuration file) that has not previously serviced a request. The next new request is sent to the next available server, and so on.

Apache Web server

Apache Web servers handle load balancing differently than the Sun Java System Web servers. Apache Web servers spawn multiple processes to handle the client load, whereas other Web servers use only a single process. Since Apache spawns multiple processes, client request loads are balanced for each Apache Web server process, not for all Apache processes. Therefore, while monitoring connection requests from an Apache Web server to multiple EAServer installations, you might initially see many new client requests from multiple Apache processes being directed to a single EAServer. In time, and as the number of client requests to the Apache Web server increases, the load is evenly distributed across all EAServer installations.

In Figure 4-2 on page 37, the Web server redirector has an entry in the redirector configuration file that directs requests for the "examples" Web application to three application servers: Jag1, Jag2, and Jag3. Requests for the Web application are distributed evenly across all three servers.

Figure 4-2: Redirector configuration file entry
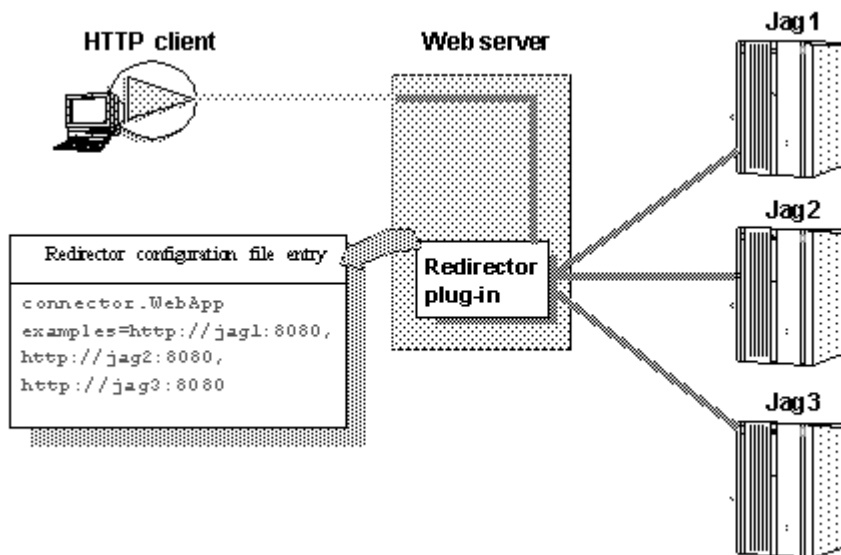
Failover and detecting unavailable application servers

The redirector provides failover for Web applications. However, to achieve transparent failover, you must mark your Web application as distributable. This allows the HttpSession object created by the Web application to be visible to other application servers.

When a request arrives from a client, the Web server redirector sends it to an EAServer application server. The redirector selects the server based on a round-robin scheme where each *new* request from a client is directed to the next available application server that can service the request. If the application server is not responding to a request from a client (for example, temporarily down for administrative reasons), the redirector marks the application server unavailable for a fixed period of time and sends the request to the next application server that can service the request.

If a request arrives from a client and an application server has been unavailable for more than the retry value, the redirector sends the request to this unavailable server. If the server responds, it is marked as available.

The default retry value is 300 seconds (five minutes). You can change the default by adding this directive to the redirector configuration file:

```
Connector.RetryAfter <seconds>
```

where *seconds* is the number of seconds the application server is unavailable.

If the redirector has connected successfully to an application server, and while awaiting the response, the network connection is broken between the redirector and EAServer, the redirector does not attempt to restore the connection. The client must retry the request.

Binding to a specific application server

If you are using the Apache or Sun Java System redirector plug-in, you can bind the redirector to a specific server. If a redirector is bound to a server and the server fails to respond to a request, the redirector returns immediately. By default, a request fails over to another server when the initial server does not respond. If you are running Web applications that are not marked as distributable, you must bind the redirector to a specific server. If a bound server fails to respond to a request, the redirector returns the error status to the client. To bind the redirector to a server, add this line to the redirector configuration file, where *serverName* is the name of the server:

```
Connector.BindToServer /serverName
```

Debugging

If you are using the Apache or Sun Java System redirector plug-in, you can trace requests by setting this directive in the redirector configuration file:

```
Connector.SessionId <ConnectorSessionId>
```

When this directive is set, the value of *ConnectorSessionId* is appended to the URL that is forwarded to EAServer. EAServer writes the URL to the server's HTTP request log, which can be helpful for debugging. For example, if you add this to the redirector configuration file:

```
Connector.SessionId ConnSID
```

EAServer writes this information to the HTTP request log:

```
10.22.85.66 - - "GET /TestHTTPS/?ConnSID=2696_000000000000 HTTP/1.0" 200 51
10.22.85.66 - - "GET /TestHTTPS/?ConnSID=2888_000000000000 HTTP/1.0" 304 0
10.22.85.66 - - "GET /TestHTTPS/?ConnSID=2889_000000000000 HTTP/1.0" 304 0
```

```
10.22.85.66 - - "GET /TestHTTPS/?ConnSID=2888_000000000001 HTTP/1.0" 304 0
10.22.85.66 - - "GET /TestHTTPS/?ConnSID=2889_000000000001 HTTP/1.0" 304 0
```

In this example, the Apache Web server process 2696 sent one request, process 2888 sent two requests, and process 2889 sent two requests. The connector's session ID is computed as:

process identifier of the Web server's process + request count

Defining a custom error page

You can specify a customized error page for the redirector to override the default message. To do so, add this setting to the redirector configuration file:

```
Connector.ErrorPage path-to-file
```

Where *path-to-file* is the full path to a text file that contains the alternate error message.

Configuring the data source for HTTPServletRequest methods

When using a Web server redirector, you can configure the source for information returned by the javax.servlet.http.HttpServletRequest methods getScheme, getServerPort, and getServerName. EAServer 5.*x* differs from EAServer 6.1 in how these properties are used. See "Proxy support," below, for more information about this as well as other related features.

See the Java Servlet 2.4 specification for additional information about the Servlet 2.4 API, as well as the user documentation of your specific development tool.

Proxy support

Proxy support is a series of properties that contain information about the machine hosting the Web redirector. Proxy support allows for scenarios where requests to the server come through a redirector. In some cases, the server constructs URLs for secondary requests, and these URLs must point to the redirector's host, protocol and port instead of the servers. Three cases exist to support this scenario:

1   Form-based authentication

2   Client-side redirection

3   Dynamically creating URLs via the getServerName(), getServerPort(), getProtocol() methods

New proxy properties and existing public APIs are used to enable proxy support.

Proxy properties

Because there is no guarantee that a Web redirector is in the same domain as EAServer, a server property called httpDomain is required in some circumstances to ensure the redirector's domain is used instead of the server's domain.

For the server to communicate with the redirector, assign these application server properties to the Web redirector:

• httpProxyProtocol – the default protocol used to connect to the Web redirector.

• httpProxyPort – the HTTP port of the redirector.

• httpsProxyPort – the HTTPS port of the redirector.

If the proxy protocol is HTTP, then httpsProxyPort is not applicable. If the proxy protocol is HTTPS, then httpProxyPort is not applicable. These properties exist for scenarios in which the server must automatically redirect, such as in form-based authentication and client-side redirects.

Public API changes

A JSP/servlet can dynamically create a URL using public methods on the HttpRequest object to get the server's port, protocol and host. When configured with a redirector, the sever's port, protocol, and host differ from the redirector. As a JSP/servlet developer, you may decide to redirect JSPs/servlets to the redirector's host/port instead of the server's. In EAServer 6.1, the server reads these HTTP header fields:

• host – to determine the host and port the client used to connect.

• sybaseredirectorheader – to determine the protocol that was used to connect to a Web redirector.

If you call any of the HttpRequest methods—getServerName, getServerPort, or getProtocol—the values are retrieved from the browser.

HTTP cookies

One of the fields of a standard cookie is called "domain." This field is checked by Web browsers to ensure that a returned cookie is from the originating domain. If it is not, the browser drops the cookie. This scenario arises when you are using the Web redirector because the server and redirector might be on different domains. For example, if the client makes a connection to the redirector on domain *a.b*, which then redirects to a server on domain *c.d*, by default the server creates a cookie with the domain set to *c.d*. When the client receives the cookie from domain *c.d*, it expects it to be from domain *a.b*, so drops it. Cookie generation uses the domain set in the httpDomain property.

Sticky sessions

When the Web redirector performs load balancing, the load is spread between a number of servers. When session data is maintained by the server, it is more efficient to send all requests for the same session to the same server (sticky sessions). To enable sticky sessions, set the includeServerInfoInSession property on the socket listener.

Installing libraries from EAServer to your Web server

Run the installation program and select the various Web redirector plug-ins to install (or select the Client Runtime option to install all of the redirectors). See Table 4-1 on page 34. Corresponding files are installed in the *WebRedirectors* subdirectory of your EAServer installation. The JAR files along with the *redirector-common-files.jar* file include the necessary files for the corresponding redirector plug-in.

When setting up a redirector for a specific Web server, unzip the appropriate redirector-specific JAR file and common file into the appropriate directory for that Web server, using WEB_SERVER_HOME as the location of the Web server software:

- Sun Java System – the *WEB_SERVER_HOME/bin/https* directory.

- Apache – the *WEB_SERVER_HOME/modules* directory.

Installing and configuring a Sun Java System Web server plug-in

This section describes how to install and configure the redirector plug-in on a Sun Java System Web server. This process requires:

1   Installing libraries from EAServer to your Web server

2   Editing the Web server configuration files.

3   Editing the redirector configuration file.

Editing the Web server configuration files

Your Web server installation includes an *obj.conf* file that contains configuration information used by the Web server. For a Sun Java System Web server, the *magnus.conf* file must include location information, and *obj.conf* must include the context paths and the EASproxy object.

Adding location information

The redirector plug-in is a Netscape Application Programming Interface (NSAPI) extension. The plug-in file name is *libjeas_httpd40.so*.

To load and initialize the plug-in, edit the *magnus.conf* file (Sun Java System), and add the locations of both the plug-in file, and the redirector configuration file, which is used by the redirector to map incoming client requests to EAServer HTTP listeners.

> **Warning!** Make a backup of *obj.conf*, and when editing the file, do not include any extra spaces, tabs, or return characters.

Add these lines at the top of *magnus.conf*:

```
Init fn="load-modules" shlib="PLUGIN_PATH" funcs="conn_init,conn_service"
Init fn="conn_init" configfile="CONFIG_FILE_PATH"
```

Where *PLUGIN_PATH* is the complete path and name of the NSAPI plug-in you are creating, and *CONFIG_FILE_PATH* is the complete path and name of the redirector configuration file. The redirector configuration file can be any name and location you choose, but must be defined by the configfile attribute.

Note Use forward slashes in the configuration file, regardless of platform.

The following sample illustrates the correct syntax for the Sun Java System Web server. The example uses "neptune" as the name of the Web server, and "conn_config" as the name of the configuration file.

Sun Java System:

```
Init fn="load-modules" shlib="WEB_SERVER_HOME/bin/https/libjeas_httpd40.so"
funcs="conn_init,conn_service"

Init fn="conn_init"
configfile="c:/webserver/https-neptune/config/conn_config"
```

Defining context paths

Context paths inform the Web server which requests are directed to the redirector. In the *obj.conf* file, search for this line:

```
<Object name=default>
```

Immediately after this line, add the lines that define the context paths. For example, if you have two Web applications named "ws_test" and "examples," the lines that direct requests of the form http://**host-name**/examples/* and http://**host-name**/ws_test/* to the redirector are:

```
NameTrans fn="assign-name" from="/ws_test/*" name="EASProxy"
NameTrans fn="assign-name" from="/examples/*" name="EASProxy"
```

Figure 4-3 illustrates how a client request for *examples/test.jsp* and *examples/images/welcome.gif* to the Web server and port Web1:8020 are redirected to jag1:8080.

Figure 4-3: "examples" requests are redirected to EAServer
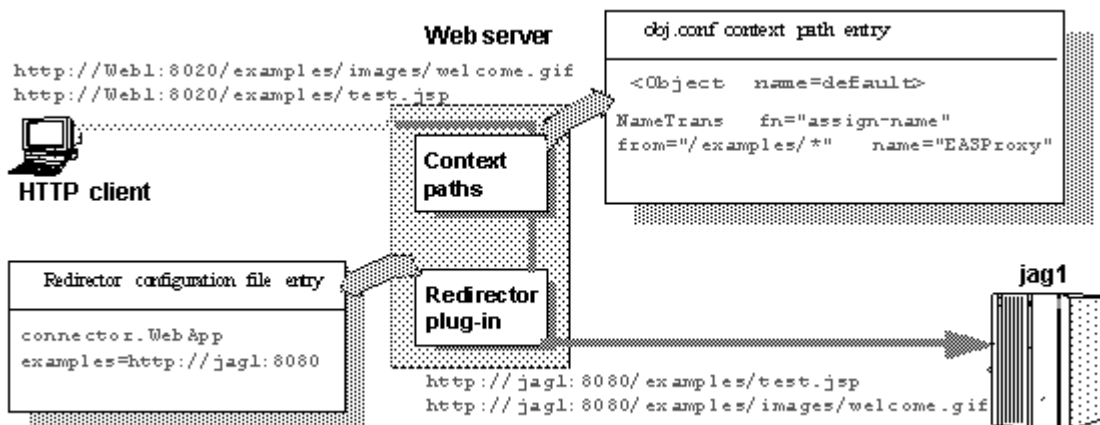
Adding the EASProxy object

At the end of the *obj.conf* file, add the lines that add the EASProxy object and specify which HTTP methods are forwarded to the redirector:

```
<Object name=EASProxy>
ObjectType fn="force-type" type="text/plain"
Service method="(GET|HEAD|POST)" fn="conn_service"
</Object>
```

Editing the redirector configuration file

This section describes how to set the redirector configuration file directives to:

• Map Web server requests to EAServer listeners

• Set the redirector logging level

Mapping Web server requests to EAServer listeners

Create and edit the redirector configuration file to add the Connector.WebApp directive, which maps requests sent to the redirector by the Web server to EAServer HTTP listeners. The syntax is:

> Connector.WebApp *contextpath1* [, *contextpath2*, and so on] =
> http://*host*:*port* [, http://*host*:*port*, and so on]

For example, if "neptune" is the name of your Web server, and
WEB_SERVER_HOME/https-neptune/config/conn_config is the name and
location of the redirector configuration file, edit *conn_config* to:

```
Connector.WebApp /testapp = http://jaghost1:8000
Connector.WebApp /estore1, /estore2, /estore3 = http://neptune:8000
Connector.WebApp /* = http://foobar:8000, http://neptune:8000
Connector.WebApp examples=http://jaghost1:8000, http://jaghost1:3080,
http://jaghost2:8000
```

- In the first line, the "testapp" Web application maps to the EAServer
  HTTP listener `http://jaghost1:8000`.

  For example, if your Web server is accepting requests on the host machine
  "web_server" on port 8020 and the client sends the request
  `http://web_server:8020/testapp/test.jsp`, the redirector forwards
  the request to `http://jaghost1:8000/testapp/test.jsp`.

- In the second line, the /estore1, /estore2, and /estore3 Web applications
  map to the EAServer HTTP listener `http://neptune:8000`.

  For example, if your Web server is accepting requests on the host machine
  "web_server" on port 8020 and the client sends the request
  `http://web_server:8020/estore1/sign_in.jsp`, the redirector
  forwards the request to `http:/neptune:8000/estore1/sign_in.jsp`.

- In the third line, "/*" matches any request that is received by the connector,
  provided the request did not match any of the other specifically configured
  Web applications.

  The redirector configuration file must contain the following mapping rule
  for each EAServer that you want to handle requests:

  > Connector.WebApp /* = http://*jaguar_host*:*jaguar_listener*

  Where *jaguar_host* is the EAServer host machine and *jaguar_listener* is
  the EAServer HTTP listener.

- The last line in the example above is a load-balancing example, where the
  "examples" Web application is available at jaghost1:8000, jaghost1:3080,
  and jaghost2:8000.

Single-line entries   You must make entries in the redirector configuration file on a single line, but
you can combine EAServer hosts in one entry or multiple entries. For example:

> Connector.WebApp /*=http://rainforest:8000,http://neptune:8000

Is the same as:

```
Connector.WebApp /*=http://rainforest:8000
Connector.WebApp /*=http://neptune:8000
```

In addition, the server hosts specified by the WebApp directive (in this case, "rainforest" and "neptune") must match the host names you define for your EAServer HTTP listeners. Use the Management Console to view and set host names for your EAServer HTTP listeners. See Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

Setting the redirector logging level

Edit the redirector configuration file to add the Connector.LogLevel directive, which sets the redirector logging level that is sent to the Web server log file. The syntax is:

Connector.LogLevel [inform | error | verbose]

Table 4-3: Logging level options

Logging level	Description
inform	Adding Connector.LogLevel inform to the redirector configuration file provides minimum logging to the Web server log file and is appropriate for production servers. This is the default logging level.
error	Adding Connector.LogLevel error to the redirector configuration file is appropriate when you want to log error messages to the Web server log file. Error messages provide additional information, such as the Web server disconnecting from EAServer or not connecting to an EAServer.
verbose	Adding Connector.LogLevel verbose to the redirector configuration file provides the highest level of logging to the Web server log file and can be used only with the EAServer debug libraries to provide additional trace information that may be useful for diagnosing problems.

Installing and configuring an Apache Web server plug-in

This section describes how to install and configure the redirector plug-in on the Apache Web server running on Windows. The process involves these steps:

1   Installing libraries from EAServer to your Web server. Your unzipped redirector file includes:

- *redirector-apache-1.3.jar* – a *libjeasproxy_mod.so* file that implements the Apache 1.3 Web server plug-in.

- *redirector-apache-2.0.jar* – a *libjeas2_mod.so* file that implements the Apache 2.0 Web server plug-in.

2   Editing the httpd.conf configuration file.

3   Editing the redirector configuration file.

For information about the Apache Web server, see Apache Core Features at http://httpd.apache.org/docs/mod/core.html.

EAPI versus non-EAPI

The redirector zip file contains the required files for the Apache plug-ins. Whether or not you are running an Extended API (EAPI) version of the Apache Web server determines which libraries you must copy to the location where the Web server normally searches for files. To determine which version of the Apache Web server is running, enter the httpd -V command. If the output from this command contains "-D EAPI," you have an EAPI version of Apache and must have the appropriate files:

- Files required for EAPI versions of the Apache Web server:

  - *libjeaseapi_mod.so*

  - *libjeasproxyeapi_mod.so*

- Files required for non-EAPI versions of the Apache Web server:

  - *libjeas_mod.so* (for Apache version 1.3) or *libjeas2_mod.so* (for Apache version 2.0)

  - *libjeasproxy_mod.so*

In addition, for EAPI versions of Apache, you must modify the LoadModule directive in the *httpd.conf* file as follows:

```
LoadModule easredirector_module libexec/libjeaseapi_mod.so
```

If you do not set this directive, the redirector may not run, and you may see a warning similar to this on the screen and in the connector log file:

```
[warn] Loaded DSO libexec/libjeas_mod.so uses plain
Apache 1.3 API, this module might crash under EAPI!
```

```
(please recompile it with -DEAPI)
```

Editing the *httpd.conf* configuration file

The *httpd.conf* file:

- Defines and loads the redirector module – the LoadModule directive informs the Apache Web server of the name and location of the redirector module to load when the Web server starts.

- Identifies the redirector configuration file – the redirector configuration file contains the mappings of incoming requests to the EAServer HTTP listener to which they are redirected.

- Defines the context paths – the LocationMatch directive defines the context paths that inform the Web server which requests are directed to the redirector. If no context path is provided, the request is serviced by the Web server.

Sample httpd.conf file   The following lines illustrate a sample *httpd.conf* file.

```
LoadModule easredirector_module
modules/libjeas2_mod.so
EASConfigFile WEB_SERVER_HOME\conf\conn_config
ServerName www.myhost.com
<LocationMatch /examples/*|/estore/* >
SetHandler eas-handler
</LocationMatch>
```

At the end of the *WEB_SERVER_HOME/conf/httpd.conf* file, add the directives defined in the sample above, where:

- LoadModule defines and loads the redirector module.

- EASConfigFile identifies the file that contains the redirector configuration parameters.

- ServerName identifies the host machine on which the Web server runs. Specify the host using either the IP address or the fully qualified domain name; for example, www.foo.com. If you do not set this directive, the redirector continues to run, but you might see this warning in the connector log file:

```
[Tue Jun 18 15:19:12 2006] [alert] httpd: Could not
determine the server's fully qualified domain name,
using 1.2.3.4 for ServerName". However the following
is fatal and the redirector will not load.
```

```
"Unable to determine host name"
```

- *conn_config* represents the redirector configuration file.

The redirector configuration file can be any name and location you choose, but it must be identified by EASConfigFile. In this example, *conn_config* is the name of the redirector configuration file. You must supply the full path to this file. See "Editing the redirector configuration file" on page 50 for a description of configuration parameters.

Warning! The Apache Web server must use the dynamic shared object (DSO) to load redirector modules and files. See your Apache Web server documentation for more information about building the Apache Web server with DSO support.

This is an Apache Web server directive:

```
<LocationMatch contextpath1 | contextpath2 | contextpath3>
SetHandler eas-handler /
</LocationMatch>
```

When an HTTP request arrives and the starting path matches *contextpath1*, *contextpath2*, or *contextpath3*, eas-handler is invoked. The *contextpath* corresponds to the Web Application context path. If the path is set to "/*", all requests are sent to the redirector. See "Defining context paths" on page 49 for more information.

Defining context paths

Context paths inform the Web server which requests are directed to the redirector. At the end of the *httpd.conf* file, add the lines that define the context paths. For example, if you have two Web applications named "ws_test" and "examples", the lines that direct requests of the form `http://host-name/examples/*` and `http://host-name/ws_test/*` to the redirector are:

```
<LocationMatch /examples/* | /ws_test/* >
SetHandler eas-handler
</LocationMatch>
```

Figure 4-4 illustrates how a client request for *examples/test.jsp* and *examples/images/welcome.gif* to the Web server and port Web1:8020 is redirected to jag1:8080.

Figure 4-4: "examples" requests are redirected to EAServer
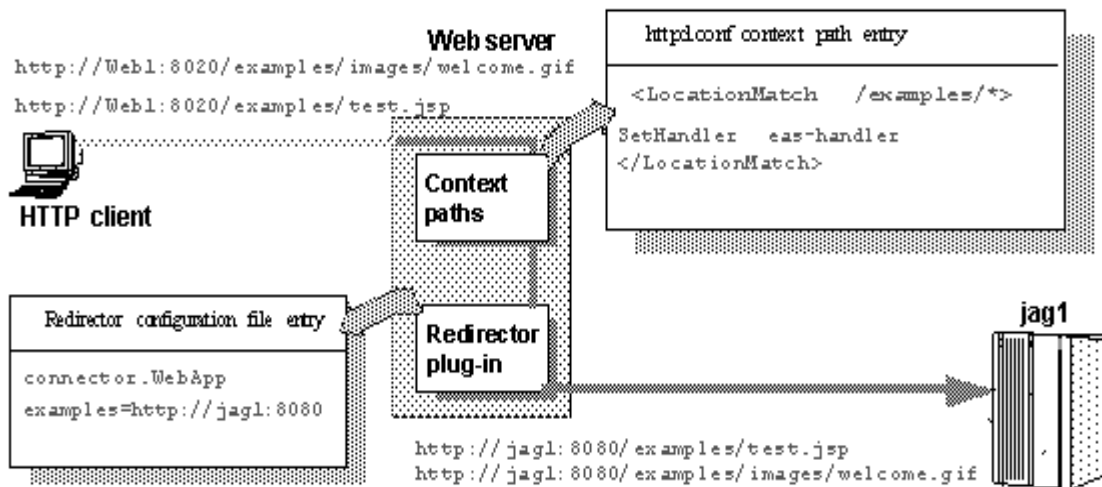
Editing the redirector configuration file

Create and configure a redirector configuration file to:

• Map Web server requests to EAServer listeners

• Set the redirector logging level

Mapping Web server requests to EAServer listeners

You must configure the redirector configuration file by adding the Connector.WebApp directive, which maps your requests to the EAServer listener to which they are redirected. The syntax is:

Connector.WebApp *contextpath1* [, *contextpath2*, and so on] = \
http://*host*:*port* [, http://*host*:*port*, and so on]

For example, if "neptune" is the name of your Web server, and *WEB_SERVER_HOME/conf/conn_config* is the name and location of the redirector configuration file that you create, edit *conn_config* to set the Connector.WebApp directive:

```
Connector.WebApp /testapp = http://jaghost1:8000
Connector.WebApp /estore1, /estore2, /estore3 = http://neptune:8000
Connector.WebApp /* = http://foobar:8000, http://neptune:8000
```

```
Connector.WebApp examples=http://jaghost1:8000, http://jaghost1:3080,
http://jaghost2:8000
```

- In the first line, the "testapp" Web application maps to the EAServer HTTP protocol listener `http://jaghost1:8000`.

  For example, if your Web server is accepting requests on the host machine "web_server" on port 8020 and the client sends the request `http://web_server:8020/testapp/test.jsp`, the redirector forwards the request to `http://jaghost1:8000/testapp/test.jsp`.

- In the second, line the /estore1, /estore2, and /estore3 Web applications map to the EAServer HTTP protocol listener `http://neptune:8000`.

  For example, if your Web server is accepting requests on the host machine "web_server" on port 8020 and the client sends the request `http://web_server:8020/estore1/sign_in.jsp`, the redirector forwards the request to `http:/neptune:8000/estore1/sign_in.jsp`.

- In the third line, "/*" matches any request that is received by the connector, provided the request did not match any of the other specifically configured Web applications. However, you must still configure the Web server to tell it which context paths should be serviced by this connector. For this mapping, you would supply the following LocationMatch directive in the *httpd.conf* file:

  ```
  <LocationMatch /* >
  SetHandler eas-handler
  </LocationMatch>
  ```

- The last line of the example above provides load balancing. The "examples" Web application is available at jaghost1:8000, jaghost1:3080 and jaghost2:8000.

Single-line entries

You must make entries in the redirector configuration file on a single line. You can combine EAServer hosts in one entry or create multiple entries; for example:

```
Connector.WebApp /*=http://rainforest:8000,http://neptune:8000
```

Is the same as:

```
Connector.WebApp /*=http://rainforest:8000
Connector.WebApp /*=http://neptune:8000
```

In addition, the server hosts specified for the WebApp directive (in this case "rainforest" and "neptune") must match the host names you define for your EAServer HTTP listeners. Use the Web Console to view and set host names for your EAServer HTTP listeners. See Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

Setting the redirector logging level

Edit the redirector configuration file to add the Connector.LogLevel directive, which determines the amount of redirector logging that is sent to the Web server log file. For a description of the logging level options, see Table 4-3 on page 46.

For example, to set the redirector logging level to "inform," add this line to the redirector configuration file:

```
Connector.LogLevel inform
```

You must also set the logging level for the Apache Web server in the *httpd.conf* file to either "info" or "error" for the redirector to produce any logging. The syntax is:

LogLevel [debug | info | notice | warn | error | crit]

For example, add this line to the *httpd.conf* file:

```
LogLevel info
```

Starting the Apache Web server

If you are starting the Web server from the command line, add *WEB_SERVER_HOME/modules* to the PATH environment variable from the command line before starting the Web server.

❖ **Verifying that the redirector plug-in works correctly**

1   Verify that all the required libraries have been copied to the *libexec* subdirectory of the Apache installation directory.

2   Verify that the *conn_config* file exists in the Apache configuration directory.

3   Verify that the Connector.WebApp directive is set correctly in the *conn_config* file. For example:

```
Connector.WebApp /* = http://hostname:8000
```

where *hostname* is the machine name where EAServer is installed, and 8000 is the EAServer HTTP port number, and "/*" appears immediately after Connector.WebApp.

4   Start the Web server. If it starts successfully, a disclaimer is displayed to the screen and written to the log file (*/testarea/apache/logs/error_log*).

5   Start EAServer on the host you specified in the *conn_config* file and from a Web browser, connect to the Web server's default page.

The redirector plug-in should redirect your request to the EAServer home page. This is the *hostname*:*port_number* set in the Connector.WebApp directive in the *conn_config* file.

Configuring EAServer to accept Web server requests

You must make sure the EAServer host and port defined in your redirector configuration file has an identical host name in the EAServer listener host and port fields. For example, this redirector entry also requires an EAServer HTTP protocol listener defined with the host name of jaghost1 and a port number of 8000:

```
Connector.WebApp /*=http://jaghost1:8000
```

See the *EAServer System Administration Guide* for instructions.

Note jaghost1 is not equivalent to jaghost.*domain*, where *domain* is the domain in which EAServer is hosted.

Configuring HTTPS support for the Web server plug-ins

You can establish HTTPS connections between your Web server redirector and EAServer on Windows machines. You should be familiar with SSL and Security Manager before establishing HTTPS connections. See Chapter 12, "Managing Keys and Certificates" in the *EAServer Security Administration and Programming Guide*.

Note Verify that your Web server redirector successfully forwards HTTP requests before you configure it for HTTPS.

HTTPS installation instructions

This section describes installation requirements for Web servers hosted on a Windows machine, including environment variable settings and which files to copy from your redirector zip file to the Web server host.

1   To redirect using HTTPS, install and extract both the *redirector-common-files.jar* file and the redirector specific files. See Table 4-1 on page 34.

2   The certificate database files are included as part of a typical installation as well as a client runtime installation.

See "Enabling HTTPS connections" to enable your Web server redirector to establish HTTPS connections with EAServer.

Enabling HTTPS connections

To enable your Web server redirector to establish HTTPS connections with EAServer:

1   Follow the same instructions as for establishing HTTP connections, for your Web server and platform.

2   Define the redirector configuration directives listed in Table 4-4. See "Editing the redirector configuration file" on page 44 (Sun Java System), or "Editing the redirector configuration file" on page 50 (Apache) for more information.

Table 4-4: HTTPS directives

Directive name	Default or valid value	Comments
Connector.Https.qop	\<sybpks_simple | sybpks_intl | sybpks_domestic | sybpks_strong>	Identifies the cipher suites the client (the Web server redirector in this case) is willing to utilize while negotiating an SSL connection.
Connector.Https.pin	\<sybase>	Specifies the PKCS #11 token PIN. This is required for logging in to PKCS #11 module for retrieving trust information.
Connector.Https.cacheSize	\<100>	SSL session IDs are cached once a successful connection has been established. When the client reconnects to the server, the client can reuse the previously established SSL session parameters by sending the old SSL Session ID to the server. This improves performance if the client is frequently connecting to, and disconnecting from, the same server. The cache size is the size of this SSL Session ID cache.
Connector.Https.SessLingerTime	\<28800> (8 hours)	The number of seconds a session ID entry is kept in the SSL session cache after the last SSL connection using that session ID is terminated.
Connector.Https.SessShareCount	\<10>	Specifies the number of concurrent SSL sessions that can use the same SSL session ID.

For a description of these directives, see CtsSecurity/SSLSessionInfo in the generated IDL documentation, which is available in HTML format in the *html/ir* subdirectory of your EAServer installation.

3   Set the JAGUAR_CLIENT_ROOT environment variable on the Web Server host to point to the location of your EAServer client installation. The directory pointed to by JAGUAR_CLIENT_ROOT must contain the *db* subdirectory, which contains the *sybcert.db* and *sybkey.db* files. The Web server process must be able to read these files. *sybcert.db* and *sybkey.db* contain certificate files, including the Jaguar User Test CA.

4   Edit the redirector configuration file to map requests sent to the redirector by the Web server to EAServer HTTPS listeners. For more information, see "Mapping Web server requests to EAServer listeners" on page 50 (Apache). Map requests to HTTPS listeners by using the appropriate *host:port* combination. See "Testing your HTTPS connections."

Testing your HTTPS connections

Once you have installed the required files and configured the Web server redirector plug-in to use HTTPS, you can test your connection to EAServer by sending HTTPS requests to a default HTTPS listener (for example, port 8001).

Port 8001 uses the Sample1 user certificate, which is signed by the Jaguar User Test CA. The Jaguar User Test CA is trusted by the Web server redirectors and installed certificates by default. Sending HTTPS requests to port 8001ensures that your Web server redirector is configured correctly. You can then define HTTPS connections for a production environment following the steps described in "Managing certificates for HTTPS connections" on page 56.

Managing certificates for HTTPS connections

To establish an HTTPS connection between your Web redirector and EAServer, you must export the CA certificate (the signing authority's certificate) that signs the user certificate for the HTTPS listener to which you want to connect. For example, if Verisign Class 1 Primary CA is the signer of the web_redirector_certificate that is assigned to an EAServer listener at port 8085, then you must install the Verisign Class 1 Primary CA certificate on your Web server host and mark it trusted for your redirector to connect to that EAServer listener.

There are several ways, depending on how you obtain and manage certificates, to export and install certificates. See Chapter 11, "Managing Keys and Certificates," in the *EAServer Security Administration and Programming Guide*.

Troubleshooting HTTPS connections

If the redirector does not start, or the Web server appears to hang, after configuring the HTTPS redirector, set the JAGUAR_RANDOMSEED environment variable to point to a file, and restart the Web server redirector. See "Setting the JAGUAR_RANDOMSEED variable" in Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

Adding FIPS and TLS support to the Web server redirector plug-ins

To enable FIPS when you are configuring the Web server, instead of using the standard redirector module, use the *_f140* version of the library:

• Apache 1.3 – *libjeasproxy_mod_f140.so*

• Apache 2.0 – *libjeas_mod2_f140.so*

• Sun Java System – *libjeas_httpd40_f140.so*

❖ **Enabling FIPS on the redirector plug-in**

1   Make changes to the corresponding configuration files. See the section that corresponds to the particular Web server, and designate the *_f140* version of the library.

2   Select a FIPS-compatible cipher suite when setting the Connector.Https.qop directive in the redirector configuration file.

3   Restart the Web server for the changes to take effect.

Managing FIPS for the redirector plug-ins

To establish FIPS connections between a redirector plug-in and EAServer, enable FIPS on EAServer. Follow the instructions in Chapter 6, "Using TLS and FIPS," in the *EAServer Security Administration and Programming Guide*.

Configuring Open Client and ODBC for EAServer

This chapter describes how to set up Open Client and ODBC drivers to work with EAServer.

Topic	Page
Overview	59
Configuring ODBC	59
Configuring Open Client	60

Overview

EAServer 6.1 can use any external Open Client that is publicly available from Sybase. Different versions of Open Client may have subtle environment differences, so refer your Open Client documentation for details. The EAServer installation includes an option to install Open Client version 12.5.1. This section describes the basic configuration tasks required to use Open Client and ODBC with the application server.

The EAServer installation includes the Merant (formerly Intersolv) DataDirect Connect ODBC Driver Manager For Sybase Databases, Version 3.60. You must select the ODBC Manager option during installation to install the ODBC Manager. EAServer does not include an ODBC driver.

Configuring ODBC

There are no ODBC drivers provided with your EAServer software. Earlier versions of EAServer included the Merant DataDirect ODBC drivers. To install and configure ODBC drivers, refer to your ODBC documentation and your database client software documentation.

If you have Merant DataDirect software, you can still use it with EAServer 6.1. Follow the Merant documentation to install and set up your Merant ODBC driver. You can set any DataDirect environment settings globally using your login shell scripts, or locally in EAServer using the *local-setenv.sh* script file. For example, a typical ODBC entry in your *$DJC_HOME/bin/local-setenv.sh* file looks like this:

```
ODBC=<path_to_your_Data_Direct_Software>
ODBCINI=$ODBC/odbc.ini
export ODBC
export ODBCINI
```

Using ODBC from other client applications

If you have an ODBC client that uses the ODBC driver manager shipped with EAServer, make sure that the client sets the ODBCINI environment variable to the *odbc.ini* file—otherwise, the driver manager gets the data source entries from the *$HOME/odbc.ini* file, if one exists.

Configuring Open Client

This section describes only the basic configuration steps to use Open Client with EAServer. For complete information on configuring Open Client, see the *Open Client/Server Configuration Guide for UNIX* on the Product Manuals Web site at http://www.sybase.com/support/manuals.

The Open Client files are installed in the *openclient-12.5.1* subdirectory of your EAServer installation if you select the Open Client option from the installer.

❖ **Configuring the Open Client environment**

1 "Set required environment variables" on page 60.

2 "Add server definitions to the interfaces file" on page 61.

3 "Verify connectivity using isql" on page 61.

Set required environment variables

EAServer 6.1 is designed to use Open Client version 12.5.1 and later. Earlier versions of Open Client use the SYBASE environment variable. More recent versions use the environment variables SYBASE and SYBASE_OCS. The default installation of Open Client 12.5.1 sets:

* `SYBASE=/work/sybase`

* `SYBASE_OCS=OCS_12-5-1`

These two variables are defined in the setup scripts for EAServer 6.1. You must ensure that your global Open Client sets them correctly.

You can also use the *local-setenv.sh* script to locally define which installation of Open Client to use.

To set the environment variables, edit the *local_setenv.sh* setup script located in *EAServer/bin*. Create this file if it does not exist. Settings in this file are added to the server environment when it starts.

After you run the setup script, set the LD_LIBRARY_PATH environment variables manually.

Prefix the Open Client *lib* and *bin* subdirectories to the LD_LIBRARY_PATH environment variable. For example:

```
setenv LD_LIBRARY_PATH $SYBASE/lib:$LD_LIBRARY_PATH
```

Add server definitions to the interfaces file

The *interfaces* file in the Open Client installation directory is a text file that defines server names and addresses. You must add entries for each EAServer or database server to which you intend to connect.

❖ **Adding server entries to the file with a text editor**

1 Create the file *$DJC_HOME/interfaces* if it does not already exist.

2 Open the file in your text editor and add an entry in this format:

```
server_name
    master tcp ether host, tds_port
    query tcp ether host, tds_port
```

where:

- *server_name* – is a logical name for the server, composed of letters, digits, and the underscore character.

- *host* – is the server machine's host name or IP address.

- *tds_port* – is the EAServer or database server's TDS listener port number.

Note The master and query lines must begin with a single tab and no spaces.

Verify connectivity using isql

Run the Open Client isql tool to verify the configuration:

```
$SYBASE/bin/isql -I $DJC_HOME/interfaces -S server_name -U user -P password
```

Where:

- *server_name* – is the server you defined in the interfaces file.

- *user* – is a valid user name for EAServer or the database server.

- *password* – is the password that accompanies the user name.

If isql shows a command prompt and not an error message, the connection is successful.

Index

V

W

Index