# SYBASE®

What's New in EAServer

## EAServer

Version 5.3

# Contents

CHAPTER 1

# New features in EAServer 5.3

These features are new in EAServer 5.3.

| Topic | Page |
|---|---|
| Merge of Axis 1.2 | 1 |
| Named classloader support | 2 |
| Oracle 10g support | 2 |
| jConnect 6.0 support | 3 |
| JDK 1.5 support | 3 |
| Unicode datatype support | 4 |
| Upgrade to Eclipse 3.0.1 | 4 |
| Upgrade to Adaptive Server Anywhere 9.0.2 (ASA 9.0.2) | 4 |

## Merge of Axis 1.2

EAServer 5.3 now supports Axis 1.2 final release. It is a Simple Object Access Protocol (SOAP) engine—a framework for constructing SOAP processors such as clients, servers, gateways, and others.

Axis makes EAServer fully compliant to Java API for XML-Based Remote Procedure Call (JAX-RPC) and SOAP with Attachment API for Java (SAAJ) from Sun.

The Axis 1.2 delivers the following key features:

- **Speed**   Axis uses Simple API for XML (SAX) parsing to achieve greater speed than earlier versions of Apache SOAP.

- **Flexibility**   The Axis architecture gives the developer complete freedom to insert extensions into the engine for custom header processing and system management.

- **Stability**   Axis defines a set of published interface, which change gradually compared to the rest of Axis.

- **Component-oriented deployment**   You can easily define reusable networks of Handlers to implement common patterns of processing your applications, or to distribute to partners.

- **Transport framework**   Axis has a clean and simple concept for designing transports, and the core of engine is completely transport-independent.

- **Web Services Description Language(WSDL)**   Axis supports the WSDL version 1.1, which allows you to easily build stubs to access remote services and also to automatically export machine-readable descriptions of your deployed services from Axis.

# Named classloader support

EAServer 5.3 now supports named classloader. The named classloader allows multiple applications to share a common classloader. It permits sharing of classes between applications without the need to be loaded by the server classloader.

Every entity that uses classloader in EAServer is allowed to define its own named classloader. The named classloader can also be configured in a heirarchy that allows a parent classloader to load shared classes.

Additionally, it reduces the memory footprint of an application since it loads the infrastructure classes just once.

# Oracle 10g support

EAServer 5.3 now supports Oracle 10g database for PowerBuilder and C/C++ components.

For more information, see "Oracle databases" in Chapter 4, "Database Access," in the *EAServer System Administration Guide*.

# jConnect 6.0 support

This new feature of EAServer is compiled using JDK 1.4, which includes the JDBC 3.0 standards and all associated interfaces. With jConnect 6.0, you can use Kerberos to authenticate Java applications without sending user names and passwords over a network.

jConnect 6.0 for JDBC provides:

- Interoperability with other JDBC drivers

- High-performance direct database access

- Zero-client installation

- Diverse and legacy database support

- Portability across any Web server and Java-enabled client

- Scalability for business-critical applications

- Security and reliability

For information about JDBC connection URLs, see "Configuring connection caches" in Chapter 4, "Database Access," in the *EAServer System Administration Guide*.

# JDK 1.5 support

EAServer 5.3 adds support for JDK 1.5 and its applications. It offers changes in the way you can output data, import, and declare classes, as well as in invoking methods.

The JDK 1.5 features focus on the following areas:

- Ease of Development

- Scalability and Performance

- Monitoring and Manageability

- Improved Desktop client

- Core XML support

- Supplementary character support

- JDBC RowSets

# Unicode datatype support

EAServer 5.3 now supports Open Client 12.5 Unicode datatype. The primary goal of the Unicode datatype feature is to make it possible to use unichar or univarchar in place of the traditional character datatype char or varchar.

This Unicode support handles the Adaptive Server Enterprise (ASE) new features and datatypes in creating a connection cache using the EAServer manager.

# Upgrade to Eclipse 3.0.1

EAServer 5.3 upgrades the current Eclipse plug-ins to support Eclipse 3.0.1. It provides the corresponding plug-ins using the Plug-in Development Environment (PDE) tool.

This tool provides a number of views and editors that makes it easier to build and manage plug-ins for Eclipse. Using PDE, you can create your plug-in manifest file (*plugin.xml*), specify your plug-in runtime, and indicate other required plug-ins. It also handles the tasks of writing the changes to the right files.

The Eclipse SDK supports the Double Byte Character Set (DBCS) locals on Windows, GIMP toolkit (GTK) and Motif Window systems, the official Chinese character set (GB18030), German, and Japanese locales.

# Upgrade to Adaptive Server Anywhere 9.0.2 (ASA 9.0.2)

EAServer 5.3 now uses ASA 9.0.2.

**Note** ASA is not an installable option and will only be installed together with the EAServer samples.

# New features in EAServer 5.2

These features are new in EAServer 5.2.

| Topic | Page |
|-------|------|
| Automatic performance tuning | 5 |
| EAServer Manager enhancements | 6 |
| In-memory mirror caching | 6 |
| FIPS and TLS support | 8 |

## Automatic performance tuning

The automatic performance tuning feature monitors EAServer response rates and modifies controllable parameters to improve these response rates.

Response rates are measured as the difference in time between an incoming request and an outgoing response for a given listener. The total time for all listeners is measured as a weighted average.

The parameters that EAServer can automatically modify control these features:

•   Connection cache size

•   Message service threads

•   Page caching

•   HTTP session replication

•   Hearbeat frequency

•   Check interval for JSPs

•   Garbage collection interval

•   CMP JDBC wrapper drivers

# EAServer Manager enhancements

In EAServer 5.2, EAServer Manager includes new configuration wizards for these entity types:

• Log Profile

• Log Category

• Log Handler

• Log Formatter

EAServer Manager launches the wizard when you create an entity of the associated type, instead of displaying a properties dialog box with default or empty values. You can also reconfigure existing entities using the wizard. To launch the wizard on an existing entity, highlight the entity's icon and choose File | Configuration Wizard. EAServer Manager continues to provide properties dialogs for all these entities. For more information on the wizard steps, see the EAServer Manager online help.

# In-memory mirror caching

EAServer 5.2 provides in-memory mirror caching for objects saved in the EAServer object cache. This allows objects to be accessed by any server in a cluster, and offers better performance than persistent database caching. For a new server that joins a mirror group or cluster, this feature provides in-memory synchronization. You can use in-memory mirror caching for stateful session components and HTTP sessions.

Mirror caching guarantees that:

• If a remote server in the current mirror group is not available, any object can still be retrieved.

• Any object in the cache is in sync with the latest copy of the object in the current mirror group, except when the originating server fails before completing a PUT operation on all servers in the group.

• A copy of an object being saved in the local cache is updated immediately on all available servers in the current mirror group, except when the originating server fails before completing a PUT operation on all servers in the group.

To use mirror caching feature, the message service must be active. The mirror object cache uses the following message queues to send and receive messages:

- <jc> – the object cache queue listens for system messages, such as server shutdown and start-up messages.

- <mc> – the mirror cache queue enables a remote server to recover lost data.

❖ **Configuring EAServer to use mirror caching for HTTP sessions**

1 Configure the server in a cluster, and designate mirror pairs:

- Chapter 6, "Clusters and Synchronization," in the *EAServer System Administration Guide* describes how to configure a cluster.

- Chapter 29, "Configuring Persistence Mechanisms," in the *EAServer Programmer's Guide* describes how to configure mirror pairs.

2 In the Web Application Properties dialog box:

a On the General tab, check the Distributable option.

b On the Advanced tab, add the com.sybase.jaguar.webapplication.distribute.type property, and set the value to "mirror."

❖ **Configuring EAServer to use mirror caching for stateful session components**

1 Configure the server in a cluster, and designate mirror pairs:

- Chapter 6, "Clusters and Synchronization," in the *EAServer System Administration Guide* describes how to configure a cluster.

- Chapter 29, "Configuring Persistence Mechanisms," in the *EAServer Programmer's Guide* describes how to configure mirror pairs.

2 In the Component Properties dialog box, set the following properties before you generate skeletons:

a On the Persistence tab, set Persistence to "Java Serialization" and set the Storage Component to "CtsComponents/JavaMirrorStorage."

b On the Transactions tab, select Automatic Failover.

c On the Instances tab, unselect Transient.

"Configuring the object cache" in Chapter 4, "EJB CMP Tuning," in the *EAServer Performance and Tuning Guide* describes how to enable and configure the object cache. The *CtsComponents/ObjectCache* component, which previously provided the default implementation of the cache manager, has been replaced by *CtsComponents/JavaObjectCache*.

# FIPS and TLS support

EAServer 5.2 supports FIPS and TLS security enhancements.

The National Institute of Standards and Technology (NIST) develops standards and guidelines for such things as security and interoperability for federal computer systems. These guidelines are called the Federal Information Processing Standards (FIPS).

EAServer uses a cryptographic module to perform encryption and decryption, signing and verification, computing a checksum (or MAC) of data, and protecting security-sensitive data. These operations are invoked by the Transport Layer Security (TLS) runtime, a software implementation of a PKCS #11 interface, and key management utility routines.

EAServer utilizes a FIPS 140-2-certified cryptographic module provided by Certicom Cryptographic libraries.

For more information, see these Web sites:

- Cryptographic Module Validation Program Web site at http://csrc.nist.gov/cryptval/ – describes the FIPS standards, contains related documents and specifications, and answers commonly asked questions.

- Certicom Security Builder Government Standard Edition (GSE) Web site at http://www.certicom.com/index.php?action=product,sbgse – describes the FIPS 140-2 certified cryptographic module that is integrated into EAServer.

- Certificate and cryptographic module information at http://csrc.nist.gov/cryptval/140-1/1401val2003.htm#351 – contains a list of certificates issued by NIST, including a copy of the Certicom certificate.

TLS is a protocol based on Secure Sockets Layer (SSL) that is used to establish secure connections between a client and server. TLS can authenticate both the client and the server, and create an encrypted connection between the two.

The TLS protocol addresses some of the security concerns of SSL v3. FIPS requires TLS for use with a FIPS cryptographic module.

See RFC 2246 at http://www.faqs.org/rfcs/rfc2246.html for a complete description of TLS.

See Chapter 9, "Using TLS and FIPS in EAServer," in the *EAServer Security Administration and Programming Guide* for complete information.

**New features in EAServer 5.1**

These features are new in EAServer 5.1.

| Topic | Page |
| --- | --- |
| EAServer Manager enhancements | 11 |
| Running components externally | 13 |
| IPV6 support | 14 |
| Server performance monitoring | 14 |
| Custom class loader enhancements | 14 |
| Systems Management | 15 |
| The EASDefault Web application | 15 |
| Jagtool enhancements | 16 |
| Alternate Solaris Java threading model | 16 |
| Private UDDI registry security enhancements | 17 |

## EAServer Manager enhancements

In EAServer 5.1, EAServer Manager includes these enhancements.

### Preferences dialog

EAServer Manager now provides a Preferences dialog, allowing you to view and edit personalized EAServer settings that are stored in your Sybase Central preferences file. To display this dialog, highlight the EAServer Manager icon in Sybase Central, and choose File | Preferences. See the online help for more information on these settings.

### New wizards

EAServer Manager includes these new configuration wizards for these entity types:

- Connection Cache Configuration

- Instance Pool

- Managed Connection Factory

- Identity

- Security Profile

EAServer Manager launches the wizard when you create an entity of the associated type, instead of displaying a properties dialog box with default or empty values. You can also reconfigure existing entities using the wizard rather than the properties dialog box. To launch the wizard on an existing entity, highlight the entity's icon and choose File | Configuration Wizard. EAServer Manager continues to provide properties dialogs for all these entities. For more information on the wizard steps, see the EAServer Manager online help.

# Log file viewer enhancements

The Log File Viewer in EAServer Manager has been enhanced to display log files of any type. You can browse the server installation to pick and view any log file, including log files written by your application. For more information, see Chapter 11, "Runtime Monitoring," in the *EAServer System Administration Guide*.

# Importing and exporting a single component

You can export and import a single component in a Jaguar JAR file. This feature allows you to copy new component definitions from the same package installed on different servers without affecting other components in the package. For more information, see "Importing and exporting components in EAServer JAR format" in Chapter 9, "Importing and Exporting Application Components," in the *EAServer System Administration Guide*.

## Web application synchronization enhancement

When synchronizing Web applications, you can optionally exclude Java class files that are listed in the Web application's custom class list. Doing so may save time if these files are already present on the host server or included in the file set of other entities that are being replicated at the same time. For more information, see Chapter 6, "Clusters and Synchronization," in the *EAServer System Administration Guide*.

## Instance pool monitoring

The runtime monitoring feature in EAServer Manager has been extended to display component instance pool statistics. For more information, see Chapter 11, "Runtime Monitoring," in the *EAServer System Administration Guide*.

# Running components externally

In EAServer 4.x and 5.0, you can run stateless C++ components in a separate process with limited access to EAServer features such as connection caching and transaction control. Running a component externally protects the server process from application problems such as memory leaks or segmentation violations.

In EAServer 5.1, stateless components of any type can run externally, with full access to server-side features such as cached connections. When you mark a component to run externally, EAServer runs it in a separate server process. You can identify which external server runs the component, and assign groups of related components to run in the same external server. EAServer starts the external server when required, and restarts the server if it stops responding.

For more information on this feature, see "Running components externally" in Chapter 4, "Defining Components," in the *EAServer Programmer's Guide*.

**Note**  The previous C++ external execution model is still supported for backward compatibility.

# IPV6 support

EAServer 5.1 supports the Internet Protocol Version 6 (IPV6) on platforms that provide the underlying network support such as Windows 2003, Windows XP, and Solaris 2.8. Windows 2000 does not support IPV6. IPV6 support also requires JDK 1.4 or a later JDK version.

By default, the server supports IPV4 connections only. To start the server with support for IPV6, specify the -IPV6 and -jdk14 command-line options. With these options, the server will accept both IPV4 and IPV6 connections. For more information, see "IPV6 support" in Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

# Server performance monitoring

EAServer 5.1 includes a new Performance Monitor feature to prevent degradation of server performance under extreme load conditions. You can configure the Performance Monitor settings to heuristically govern the processing requests to prevent performance degradation due to overuse of available resources. For details, see Chapter 9, "Using the Performance Monitor," in the *EAServer Performance and Tuning Guide*.

# Custom class loader enhancements

EAServer 5.1 provides a new version of the custom Java class loader that allows hot refresh of Java classes without restarting the server. The version 2 class loader offers improved diagnostics and allows you to configure the delegation order for parent-child class loader relationships.

For compatibility with applications configured for earlier EAServer versions, EAServer uses the previous class loader version (version 1) by default. To begin using the new class loader, configure the class loader version in the EAServer Manager Server Properties dialog box. On the Java VM tab, select Use Jaguar Class Loader Version 2.

For more information, see Chapter 30, "Configuring Custom Java Class Lists," in the *EAServer Programmer's Guide*.

# Systems Management

EAServer 5.1 includes support for the Simple Network Management Protocol (SNMP), which includes an SNMP master agent from J.AgentX. This replaces the functionality provided previously by Java AgentX.

The graphical user interface has also been updated. The Web Console plug-in allows you to browse the MBeans and services installed on either your local server or a remote server using a Web browser. You can add and modify MBeans and services using wizards, rather than by editing XML. The console also allows you to set properties and invoke methods on the MBeans, and start, stop, restart, and refresh services. For more information, see Chapter 13, "Using Systems Management," in the *EAServer System Administration Guide*.

# The EASDefault Web application

In EAServer 5.1, all Web pages that are not part of a Web application are added automatically to the EASDefault Web application. You configure the properties for EASDefault the same as you do for other Web applications.

Initially, the context path for EASDefault is "/", and the *WEB-INF* directory is created under *$JAGUAR/html*. When users access http://*host*:*port*/, EAServer invokes the welcome page of the Web application whose context path is "/". Because the EASDefault welcome page is *index.html* and the context path is "/", EAServer invokes *$JAGUAR/html/index.html*. To change this behavior, you can either:

*   Change the EASDefault welcome page, or

*   Change another Web application's context path to "/" to display its welcome page.

To configure most server-level properties for servlets and JSPs, you use the Web Application Properties dialog box, instead of the Server Properties dialog box. For details, see Chapter 21, "Creating Web Applications," and Chapter 22, "Creating Java Servlets," in the *EAServer Programmer's Guide*.

# Jagtool enhancements

In EAServer 5.1, jagtool includes the following enhancements. For information on jagtool, see Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

## Web application synchronization enhancement

When synchronizing Web applications with the sync command, you can optionally exclude Java class files that are listed in the Web application's custom class list by adding `-syncwebappjavaclasses false` to the command. Doing so may save time if these files are already present on the host server or included in the file set of other entities that are being replicated at the same time.

## New flushstaticpage command

The new flushstaticpage command allows you to flush the static page cache from the command line or Ant build scripts. For more information, see the flushstaticpage reference page in Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

# Alternate Solaris Java threading model

On Solaris 2.8, you can specify the serverstart.sh option `-altthrdlib`. Specifying this option causes EAServer to run with the alternate JDK thread library that uses one-to-one mapping between kernel threads and Java threads. This threading model can yield better performance and reliability than the default many-to-many threading model. For serverstart.sh syntax, see "Starting the server" in Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

# Private UDDI registry security enhancements

EAServer 5.1 includes security enhancements for the private UDDI registry, including three new roles and security constraint support. For more information, see "Controlling access to resources" in Chapter 7, "The Private UDDI Server,"in the *Web Services Toolkit User's Guide*.

**New features in EAServer 5.0**

These features are new in EAServer 5.0.

## EAServer Manager enhancements

Jaguar Manager has been renamed EAServer Manager, and contains these usability enhancements:

- Support for simultaneous connections to multiple servers. You can manage multiple server sessions simultaneously. Each open connection is represented by an icon in the Sybase central tree view. In addition, server-side SSL certificates are now managed from the Certificates folder in the EAServer Manager tree view, rather than using a separate Sybase Central plug-in.

- Support for secure connections to the server being managed. You can enable SSL to secure the connection between EAServer Manager and the server.

- Online help. The online help for EAServer Manager now includes the EAServer product manuals in addition to the context-sensitive help. To access the help, click Help if enabled in any dialog box, or choose Tools | EAServer Manager Help.

- More wizards. Several new wizards are available, as listed under "New wizards" on page 20.

- Undeployment support. When deleting entities, you can choose whether to delete only the entity properties files (as done in earlier EAServer versions) or to delete all files generated when the entity was deployed. See "Undeployment of packages, Web applications, and applications" on page 22 for more information.

- US Government Section 508 compliance. EAServer Manager and Security Manager comply with the accessibility requirements of Section 508. You can use screen readers and accessibility aids such as:

  - The JAWS screen reader from Freedom Scientific at http://www.freedomscientific.com/fs_products/software_jawsinfo.asp.

  - The Java Access Bridge, available from Sun's Web site at http://java.sun.com/products/accessbridge/. The Java Access Bridge is required to use JAWS with Java user interfaces such as Sybase Central.

  Also, both of these tools support working without a pointing device—for details, see "Keyboard navigation" in Chapter 2, "Sybase Central Overview," of the *EAServer System Administration Guide*.

- Server diagnostics export. You can export a diagnostic log file that describes the server version and environment in detail. To use this feature, highlight the server icon, choose Export Diagnostic Log, and enter a file name.

## New wizards

EAServer Manager includes these new wizards:

- **Server Configuration and New Server wizards**   These wizards walk you through the configuration of new or existing servers. To run the New Server wizard, highlight the top-level Servers folder, then choose File | New Server Wizard. To run the Server Configuration wizard, highlight the server icon, then choose File | Server Configuration Wizard.

- **Performance Tuning wizards**   These wizards guide you through the tuning of server, component, Web application, servlet/JSP or server settings that affect performance. To run them, highlight the server, component, Web application, or JSP icon, then choose File | Performance Tuning Wizard.

- **Debug Settings wizards**   These wizards step through the configuration of settings that enable tracing and debugging of components and servers. To run the wizard on a server, highlight the server icon, then choose File | Server Debug Settings Wizard. To run the wizard on a component, highlight the component icon, then choose File | Component Debug Settings Wizard.

- **Web Application Security Configuration wizard**   This wizard walks you through the settings that control the Web application authentication mechanisms, access restrictions, and other security properties. To start the wizard, highlight the Web application's icon and choose File | Security Configuration Wizard.

- **CMP Configuration wizard**   This wizard steps through the configuration of an EJB entity bean component that uses Container Managed Persistence (CMP). To run the wizard, highlight the component icon and choose File | CMP Configuration Wizard.

- **Cluster Configuration and New Cluster wizards**   These wizards walk you through the configuration of new or existing clusters. To run the New Cluster Wizard, highlight the top-level Clusters folder, then choose File | New Cluster Wizard. To run the Cluster Configuration wizard, highlight the cluster icon, then choose File | Cluster Configuration Wizard.

- **J2EE Deployment Repair wizards**   These wizards help you find and correct common problems in EJB components, J2EE connectors, and Web applications that have been deployed from J2EE archives. For example, EJB components may use security role names that have not been mapped to EAServer role names or JDBC resource references that have not been linked to EAServer connection caches. You can run these wizards on EJB components, packages containing EJB components, Web applications, J2EE connectors, and applications. To run the wizard, highlight the entity's icon, and choose File | *entity* Configuration, where entity is Component, Package, or so forth. If this menu item is disabled (dimmed), EAServer has found no problems in the entity configuration.

## Undeployment of packages, Web applications, and applications

When deleting a package, Web application, or application, you can choose whether to delete only the entity's properties file, or to delete all files that have been generated by the deployment of the entity, including component stubs and skeletons and IDL interface and datatype definitions. By default, EAServer Manager performs a full deletion. In earlier EAServer versions, only properties files were deleted. When deleting with jagtool, add the type option to the delete command to specify whether to use full or simple deletion

This feature is useful for deleting entities that have been imported from a J2EE archive file. Full deletion makes removing the entity as easily as the original deployment.

# HTTP directory browsing

EAServer 5.0 includes built-in support for HTTP directory browsing. When a Web client submits a request that ends in a directory name, and the directory does not contain a Welcome file, EAServer returns a listing for the directory. You can configure which directories can be browsed. For more information on this feature, see "HTTP Directory Browsing" in Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

# EAServer Transaction Manager

EAServer 5.0 includes the EAServer Transaction Manager, which supports the specifications for the Java Transaction API (JTA) 1.0, and complies with the JTS and the Object Transaction Service (OTS) and X/Open Architecture (XA) standards. The JTS transaction coordinator integrates the functionality of shared connections, OTS/XA, and JTS/JTA transaction modes, and uses two-phase commit to coordinate transactions among multiple databases.

The EAServer Transaction Manager includes:

• Resource recovery and transaction logging

• Transaction interoperability

• Resource manager

The EAServer Transaction Manager enables EAServer to control the scope and duration of transactions across multiple resource managers. It also provides the ability to synchronize transactions and to communicate with other transaction managers using CORBA OTS.

For more information about this feature, see Chapter 2, "Understanding Transactions and Component Lifecycles," in the *EAServer Programmer's Guide*.

# Java Connection Management enhancements

The JCM_WAIT, JCM_NOWAIT, and JCM_FORCE flags, which in earlier EAServer versions controlled connection allocation, are no longer used by the JCMCache.getConnection(int) method. You can achieve the same functionality by calling getConnection(0), and configuring the connection cache properties in Table 4-1, using the Caching tab in the Connection Cache Properties dialog box in EAServer Manager.

*Table 4-1: Properties to configure connection cache size*

| Property | Description |
|---|---|
| Minimum Connection Pool Size | The minimum number of pooled connections, allocated at server start-up. If not set, the default is 0. |
| | With jagtool, you can set as com.sybase.jaguar.conncache.poolsize.min. |
| Maximum Connections | The absolute maximum number of connections that can be created from the cache. Requests for excess connections either block or fail. A value of 0 indicates that there is no limit. |
| | With jagtool, you can set as com.sybase.jaguar.conncache.poolmanager.maxconnection. |
| Maximum Connection Pool Sizes | The maximum number of connections that can be cached. If connections are allocated beyond this number, the cache manager drops the excess connections when they are released. |
| | With jagtool, you can set as com.sybase.jaguar.conncache.poolsize.max. |
| Wait for Connections | When the maximum connections limit is reached, specifies whether requests for excess connections fail immediately or wait until a connection is released. If this setting is enabled, the request waits. |
| | With jagtool, you can set as com.sybase.jaguar.conncache.wait. |

For information on tuning these settings, see Chapter 6, "Database Access Tuning," in the *EAServer Performance and Tuning Guide*.

Table 4-2 describes three possibilities for configuring connection cache properties to duplicate the functionality that was previously available by passing the JCM flags to JCMCache.getConnection.

*Table 4-2: Configuring connection caches*

| If components previously passed this flag to getConnection | Configure these connection cache property values |
|---|---|
| JCM_WAIT | Set the value of com.sybase.jaguar.conncache.poolmanager.maxconnections equal to the value of com.sybase.jaguar.conncache.poolsize.max, and set the value of com.sybase.jaguar.conncache.poolmanager.wait to true. |
| JCM_FORCE | Set the value of com.sybase.jaguar.conncache.poolmanager.maxconnections to 0. |
| Combination of the two | For easy access to resource connections, set the value of com.sybase.jaguar.conncache.poolmanager.maxconnections to 0. |

For more information about these properties, see Appendix B, "Repository Properties Reference," in the *EAServer System Administration Guide*.

# Enhanced systems management support

EAServer 5.0 includes enhanced support for remote systems management. The implementation is based on the Java Management Extensions (JMX) agent management framework. It provides the following enhancements:

- Allows you to create management beans (MBeans) that can be run in a JMX framework.

- Supports both SNMP and JMX management tools.

- Allows you to monitor servers, clusters, and key server subsystems such as the message service, the component dispatcher, and network listeners.

For more information on this feature, see Chapter 13, "Using Systems Management," in the *EAServer System Administration Guide*.

# Enhanced Solaris C++ compiler support

The Solaris version of EAServer 5.0 supports 4.*x* and 6.*x* versions of the Solaris C++ compiler. The binary format of executable files and libraries differs between the two compiler versions, so you must choose one version for all C++ components running in a server.

When starting the server, specify the compiler compatibility version with the -compilerversion option. For example, for compatibility with the 6.*x* compiler:

```
serverstart.sh -compilerversion 6x
```

Or for compatibility with the version 4.*x* compiler:

```
serverstart.sh -compilerversion 4x
```

If you run an EAServer cluster, all servers in the cluster must use the same compiler version.

When linking component shared libraries or client binaries, you must link with the EAServer libraries that match your compiler version. Choose the appropriate directory from those listed below:

*   *lib* contains libraries that are compatible with the 6.*x* compiler, stripped of symbol information for production use.

*   *devlib* contains libraries that are compatible with the 6.*x* compiler, for debugging use.

*   *lib_sol4x* contains libraries that are compatible with the 4.*x* compiler, for production use.

*   *devlib_sol4x* contains libraries that are compatible with the 4.*x* compiler, for debugging use.

The generated make files for C++ components link with 6.*x* libraries by default. To use 4.*x* libraries, edit the definition of the LIB macro in the makefile, and change the paths to the library directories.

When running EAServer clients that are compiled from C++, set the LD_LIBRARY_PATH variable to include libraries that are compatible with the compiler version used to create the client executable.

# Server logging enhancements

EAServer 5.0 includes a more configurable logging mechanism that allows integration with the JDK 1.4 Java Logging package or the Apache Log4j logging system. A server's logging properties are defined in a **Log Profile**, which defines the logging subsystem used as well as other properties, such as output destinations, formats, and the level of severity required before a message is recorded. You can also configure different log profiles for the debug and production server versions.

You can use the following logging subsystems:

- The built-in EAS subsystem, which offers the same functionality available in EAServer 4.x versions, plus several enhancements:

  - The ability to configure log levels. so that messages below a specified level of severity are discarded.

  - Support for different logging configurations in the debug and production servers.

  - Optional archiving and compression of previous log file versions.

  - More control over message formatting.

- Apache Log4j, which is commonly used on large projects. For more information, see the Apache Log4j Documentation at http://jakarta.apache.org/log4j/docs/api/overview-summary.html.

- The Java Logging package, included in JDK 1.4. This API is Sun's proposed standard for logging in Java applications. For more information, see the Java Logging documentation at http://java.sun.com/j2se/1.4.1/docs/guide/util/logging/overview.html. To use this package, your server must be running JDK 1.4 or a later JDK version.

If you use the Log4j or Java Logging packages, you can extend default behavior by plugging in your own code that implements the required interfaces. For example, you can install Log4j log handler classes that write messages to the Windows NT event log or to a database. Also, if you use one of these packages to log messages from your own component or application code, you can configure the server's log profile so that server log messages go to the same destinations.

## Logging APIs

Regardless of the logging system you use, you can write messages to the log using all of the methods supported in earlier versions of EAServer, such as:

*   System.out.println or Jaguar.writeLog from Java code running in the server

*   ErrorLogging.log from PowerBuilder® NVO components

*   JagLog from C or C++ components

*   IJagServer.writeLog from ActiveX components

In addition, if you use Log4j or the Java Logging system, you can log messages from in-server Java code by calling the logging API directly.

## Managing log profiles

"Configuring log profiles" in the *EAServer System Administration Guide* describes how to manage log profiles.

# Runtime monitoring enhancements

EAServer 5.0 includes these new runtime monitoring features:

*   You can trace the SQL commands sent through connection caches. For more information, see "SQL tracing properties" in Chapter 4, "Database Access," in the *EAServer System Administration Guide*.

*   In EAServer Manager, HTTP and IIOP network monitoring shows thread usage, and IIOP network monitoring shows the host and thread information for current clients. For information on runtime monitoring with EAServer Manager, see Chapter 11, "Runtime Monitoring," in the *EAServer System Administration Guide*.

*   The monitoring APIs have been enhanced as follows:

    *   Jaguar::Monitoring supports new keys to retrieve the configured values for maximum HTTP threads and the maximum number of simultaneous client threads, as well as the existing keys for current thread usage.

    *   Jaguar::Monitoring includes a new method, getConnectedUsers, that returns information on IIOP clients.

• Jaguar::PerfMonitor is a new performance monitoring interface that provides performance statistics in a per-second, per-minute, and per-hour bucket model.

• Jaguar::StatProvider and Jaguar::StatProviderController are new interfaces implemented by *statistic provider* components that collect performance statistics. EAServer 5.0 includes statistics providers for the connection caching and HTTP protocol handler subsystems. You can also implement your own statistics providers using these interfaces.

For information on the monitoring APIs, see the generated HTML documentation for the Jaguar IDL module, in this file within your installation:

```
html/ir/Jaguar.html
```

• New jagtool commands getserverinfo and getservicestate allow you to get additional information about the server status. getserverinfo returns the server status and version number. getservicestatequeries the state of service components. This command is useful in scripts that start or restart servers; you can use it to determine whether the server is ready to accept client connections by checking whether the name service status is "STOPPED." Custom services can implement an additional method, getServiceState, to allow jagtool to query their status. For more information, see these chapters:

• Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

• Chapter 33, "Creating Service Components," in the *EAServer Programmer's Guide*.

# Admin role granularity

In EAServer version 5.0, this feature allows finer control over users' permissions within EAServer. Members of the Admin role can use EAServer Manager or jagtool to grant permission to perform specific actions on EAServer entities to other roles. These actions include:

• Restarting and refreshing servers

• Shutting down servers

•   Creating, updating, or deleting applications, packages, servlets, and Web applications

For more information, see Chapter 13, "Security Configuration Tasks," in the *EAServer Security Administration and Programming Guide*.

# Web Services Toolkit

Web Services Toolkit (WST) combines the architecture and tools required for creating, managing, and administering Web services contained in EAServer, and has many enhancements to WST version 4.x, including:

•   A private UDDI server

•   GUI based development and management platform

•   Web based administration console

•   Additional command line tool support for wstool and wstkeytool

See Chapter 1, "Overview of Web Services in EAServer," of the *Web Services Toolkit User's Guide* for more information.

# jagtool enhancements

EAServer 5.0 includes these enhancements to jagtool:

•   The new command getservicestate allows you to query the state of service components. This command is useful in scripts that start or restart servers, because you can use it to determine whether the server is ready to be configured.

•   The new command grantroleauth allows members of the Admin role to grant permission to perform specific actions on a given entity to a given role. If the entity is a server, members of the role can be granted permission to restart, refresh, or shutdown the server. If the entity is an application, Web application, servlet, or package, members of the role can be granted access to those resources. The inverse command removeroleauth removes permissions to perform specific actions from a given role.

- The new command `rebind` allows you to rebind the name servers in a cluster.

- The `ejbref` command has been enhanced to support EJB local references.

- The `export` command has been enhanced to take an `emptycachetags` argument. If your Web applications use the partial page caching tag library as described in "Partial page caching" on page 31, this option allows you to export an empty implementation of the tag library to allow portability to other J2EE application servers that do not support this tag library.

For descriptions of these commands, see Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

# PowerDynamo Web site conversion utility

EAServer 5.0 includes the Dyn2JSP utility to convert PowerDynamo™ Web sites into JSP-based J2EE Web applications. For information on using the tool, see the HTML documentation included in the *PDynamo2JSP* directory of the installation.

# Servlet response caching improvements

Internally, the servlet page caching mechanism has been enhanced to provide better performance. There are no changes to the user interface. Additionally, you can cache parts of a response individually using the Partial page caching feature described below.

# Partial page caching

Partial page caching allows you to cache parts of a response. This mechanism is useful when pages contain volatile content, such as calculation results, but otherwise have static content such as headers and footers. Partial page caching is supported by a tag library for use in JSPs, and a public API for use in servlets. For more information, see "Using partial page caching" in Chapter 5, "Web Application Tuning," in the *EAServer Performance and Tuning Guide*.

CHAPTER 5

# New Features in EAServer Versions 4.2.3, 4.2.2, 4.2.1, and 4.2

# New features in version 4.2.3

EAServer 4.2.3 includes these new features:

- Server stack size

- Sharing of JSP class files

- Suppressing SSL error messages

- Servlet authenticated session enhancement

- Jagtool enhancements

- Web server redirector enhancements

## Server stack size

On Windows, the default per-thread stack size for the server executable has changed to 256K (previously 512K). The smaller setting makes more memory available for other uses. For more details on this setting, see Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

## Sharing of JSP class files

You can configure Web applications to allow sharing of JSP class files among different servers that run from the same EAServer installation. To enable this feature, set the Web application property com.sybase.jaguar.webapplication.sharecompiledjspclasses to true. With this setting in effect, EAServer compiles and loads JSP classes from the directory:

Repository/WebApplication/*WebAppName*/WEB-INF/classes

Where *WebAppName* is the Web application name.

## Suppressing SSL error messages

EAServer 4.2.3 includes new settings which allow you to suppress client SSL errors from the server log file. In EAServer Manager, the Log SSL Error setting for listener and security profile properties specifies whether EAServer logs SSL errors from client connections to the listener. To suppress logging of SSL errors, unselect this option for each SSL listener and the security profile used by the listener.

If using jagtool, these settings correspond to the listener property com.sybase.jaguar.listener.logsslerr and the security profile property com.sybase.jaguar.security.logsslerr. The default for both is true.

## Servlet authenticated session enhancement

EAServer 4.2.3 includes a new server property to further protect against replay attacks against secure Web applications. The server property com.sybase.jaguar.server.servlet.serverCheckPeerIPForHttpSession can be true or false; the default is false. When true, EAServer binds the client IP address to each servlet authenticated session and validates subsequent requests to insure the IP address is the same.

## Jagtool enhancements

For information on jagtool, see Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*. These enhancements are new in EAServer 4.2.3.

### Installing and removing service components

The install and remove commands have been enhanced to support service components. You can install service components to a server with this syntax:

    jagtool install Service:*package*/*comp* Server:*server*

You can remove service components from a server with this syntax:

    jagtool remove Service:*package*/*comp* Server:*server*

Where:

• *package* is the package name

• *component* is the component name

• *server* is the server name

## Retrieving runtime monitoring statistics

The new command getmonitorstats retrieves and prints runtime monitoring statistics from the server that you are connected to. For details, see the getmonitorstats reference page in Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

## New jmsmanage_selectors command

The new jmsmanage_selectors command allows you to add and remove selectors to and from JMS message queues. For details, see the jmsmanage_selectors reference page in Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

## New jmsmanage_listeners command

The new jmsmanage_listeners command allows you to add and remove listeners to and from JMS message queues. For details, see the jmsmanage_listeners reference page in Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

# Web server redirector enhancements

The following Web server redirector plug-in enhancements are new in EAServer 4.2.3.

## Support for Netscape Enterprise Server 6.1

EAServer version 4.2.3 includes a Web server redirector plug-in for Netscape Enterprise Server (NES) version 6.1 on platforms that support this Web server. For more information, see the EAServer *Installation Guide* and *Release Bulletin* for your platform.

## New Web application property

When using a Web server redirector, you can configure the source for information returned by the HTTPServletRequest methods getScheme, getServerPort, and getServerName. To configure the source, set the Web application property com.sybase.jaguar.webapplication.get-serverinfo-from. Set this property for each Web application used by your application, choosing a value from the following table:

| Value | Specifies |
|-------|-----------|
| source | Return the server, host and scheme (protocol) for the web server that is running the redirector. |
| server | Return the server, host, and scheme (protocol) for the EAServer listener that the redirector connects to. |
| proxy | Return the scheme (protocol), host and port from the HTTP proxy settings on the HTTP Config tab in Server Properties. If these settings are not present, use the EAServer listener values. |

## Custom error page for redirector

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.

# New features in version 4.2.2

EAServer 4.2.2 includes these new features:

- Netegrity SiteMinder Integration

- Custom security component API changes

- Changing the effective user ID for the server process

- Multiple listener hosts

- Standalone jagtool

- Enhanced Windows service support

- Adaptive Server Enterprise wide table support

- Enhanced character set support

- Apache Web server redirector supports EAPI

- New EJB CMP component properties

- EJB session bean passivation

- Message service enhancements

## Netegrity SiteMinder Integration

EAServer 4.2.2 supports integration with Netegrity SiteMinder security software. Netegrity SiteMinder provides single sign-on and centralized management of Web, database, and software resources in enterprise applications. For more information, see "Netegrity SiteMinder Integration" in Chapter 10, "Creating and Using Custom Security Components," in the *EAServer Security Administration and Programming Guide*.

## Custom security component API changes

EAServer 4.2.2 augments the APIs available to create custom authentication services with the new features described in this section. You can create custom authentication components using the J2EE standard Java Authentication and Authorization Service (JAAS) API or the EAServer CtsSecurity interfaces. For more information, see these chapters in the *EAServer Security Administration and Programming Guide*:

- Chapter 10, "Creating and Using Custom Security Components"

- Chapter 11, "Using the JAAS API"

### Support for multiple JAAS configuration file sections

Beginning in EAServer 4.2.2, you can identify a configuration file section by setting the server property com.sybase.jaguar.server.jaas.section to the name of the configuration file section to use for the server. If you do not set this property, the default is the name of the server. Setting this property allows you to use the same configuration section for multiple servers.

## Retrieving additional user session details in a JAAS login module

Beginning in version 4.2.2, EAServer allows you to retrieve additional IIOP or HTTP user session information when using the JAAS API to install custom security implementations.For more information, see "Retrieving additional user session details in a JAAS login module" in Chapter 11, "Using the JAAS API," in the *EAServer Security Administration and Programming Guide*.

## Retrieving HTTP session information in a custom authentication service

You can also retrieve additional client session information in custom authentication services that implement the CtsSecurity::AuthService API. In a custom authentication component implemented in Java, you can call the com.sybase.jaguar.server.Jaguar.getHttpServletRequest() method to retrieve the HTTP servlet request (if any) that triggered the authentication event. This method returns null if the authentication event is not associated with an HTTP request (for example, if the authentication is for a component invocation).

## Supporting external single sign-on providers

EAServer 4.2.2 allows integration with external single sign-on authentication software such as Netegrity SiteMinder. EAServer 4.2.2 includes custom security components to support Netegrity, and you can implement support for other services by implementing your own custom security components.

A new API, CtsSecurity::CallerPrincipalService allows you to implement a component that tells EAServer the effective user ID when authentication occurs outside of EAServer. For details on creating and installing a caller principal service, see the HTML reference documentation for this interface in file *html/ir/CtsSecurity.html* in your EAServer installation directory.

The com.sybase.jaguar.server.http.sso server property specifies whether sign-on occurs externally. Set this property to true if you are using an external single-sign on provider.

## Changing the effective user ID for the server process

Beginning in version 4.2.2, EAServer on UNIX platforms allows you to configure an effective user name and group for the server to run as. This feature is useful if you start the server while logged in as a UNIX user with administrator privileges: you can start the server with administrator privileges, but the server switches to an account that has fewer privileges before it begins accepting client connections. For more information, see "Changing the effective user ID of the server process" in Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

## Multiple listener hosts

Beginning in version 4.2.2, EAServer allows you to configure multiple host addresses for network listeners. This feature is useful when the machine is configured with multiple network interfaces. When specifying the listener host, you can enter multiple host names or IP addresses, separated by commas. You can also enter the special value 0.0.0.0, which causes EAServer to listen on all of the machine's host or IP addresses. When using multiple host addresses, EAServer creates a listener for each host on the specified port. For more information on configuring listeners, see Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

## Standalone jagtool

jagtool and jagant now support a standalone command mode, which allows you to configure an EAServer installation without requiring a connection to a server. This enhancement is helpful in situations where it is not convenient to start a server, for example, when using jagtool to configure new installations. For more information, see Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

## Enhanced Windows service support

On Microsoft Windows platforms, EAServer versions earlier than 4.2.2 allow you to run EAServer as a Windows service so that your application server starts automatically when the system boots. EAServer 4.2.2 enhances this functionality by allowing you to run multiple servers from different installations as Windows services.

You can specify the service name when installing a server as a service or manually starting or stopping the service. To specify the service name, use the -servicename argument in addition to the arguments documented in Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*. For example, this command installs the Jaguar server as a Windows service named MyJaguarService:

```
serverstart -servername Jaguar -servicename MyJaguarService -install
```

If you do not specify a service name, the default is the server name.

EAServer 4.2.2 removes the requirement that the JAGUAR and PATH variables be configured as system environment variables for servers that run as services.

## Adaptive Server Enterprise wide table support

In EAServer 4.2.2, the EAServer version of Open Client™ Client-Library™ adds support for wide tables (varchar/varbinary columns more than 255 bytes long and tables with more than 255 columns) when connected to Adaptive Server® Enterprise version 12.5 or later.

## Enhanced character set support

EAServer 4.2.2 supports several new character sets. For the full list, list the subdirectories of the *charsets* directory. Each subdirectory matches the name of a supported character set.

## Apache Web server redirector supports EAPI

On Windows platforms, EAServer 4.2.2 adds support for running the Web server redirector with Apache 1.3 compiled with the Extended API (EAPI) option. For more information, see the *EAServer Installation Guide*.

# New EJB CMP component properties

Table 5-1 lists new component properties that you can set on the Advanced tab in the EAServer Manager Component Properties dialog box for EJB entity beans that use Container Managed Persistence (CMP). Actual property names are prefixed with com.sybase.jaguar.component. These properties were added for the 4.1.3 and 4.2.2 releases, as indicated in the table.

*Table 5-1: New component properties*

| Property | Description |
|---|---|
| com.sybase.jaguar.component.lwc (New in 4.1.3) | Enables the EJB lightweight container. |
| com.sybase.jaguar.component.cmp_iso_level (New in 4.1.3) | Specifies the effective transaction isolation level for EJB CMP entity beans. |
| com.sybase.jaguar.component.defer (New in 4.1.3) | When using one of the Sybase CMP wrapper drivers, specifies which SQL commands may be deferred to the end of the transaction. |
| com.sybase.jaguar.component.selectForUpdate (New in 4.1.3) | Similar to the Select With Lock (selectWithLock) Persistence setting, but requests an exclusive database lock be obtained at select time to avoid deadlocks during lock promotion. If this is used, also consider configuring the database table for row-level locking. |
| com.sybase.jaguar.component.tlc.sort (New in 4.1.3) | Specifies that transaction local cache entries for this component should be sorted before ejbStore is called. Setting this property to true helps to avoid deadlock when separate transactions concurrently update multiple instances of the same component. You cannot enable this unless the primary key class implements the java.lang.Comparable interface. Most java.lang utility classes implement this interface, such as String, Integer and so forth. |
| com.sybase.jaguar.component.softLock (New in 4.2.2) | See "Soft locking" on page 44. |
| com.sybase.jaguar.component.softLock.timeout (New in 4.2.2) | See "Soft locking" on page 44. |
| com.sybase.jaguar.component.touchColumn (New in 4.2.2) | Used when the selectForUpdate property is true. For databases such as Sybase Adaptive Server Enterprise that do not support select for update locking syntax, EAServer locks rows by issuing a no-change update statement. This property specifies which column to update. |

| Property | Description |
|---|---|
| com.sybase.jaguar.component.ts.triggers<br><br>(New in 4.1.3) | Specifies that the storage component (CtsComponents/JdbcStorage) should automatically create database triggers to enforce the required timestamp semantics when using table-level timestamps. |
| com.sybase.jaguar.component.tx_retry<br><br>(New in 4.1.3) | Enables automatic transaction retry for EJB CMP entity beans that use optimistic concurrency control (OCC). |

**Enabling the lightweight container**

Beginning in EAServer 4.2.2, you must explicitly enable the lightweight EJB container (LWC) for each component that uses it. See "Lightweight EJB container" on page 49 for more information.

## CMP isolation level

Beginning in EAServer 4.1.3, you can set the property com.sybase.jaguar.component.cmp_iso_level to specify the effective transaction isolation level for CMP entity beans. This setting allows the performance benefits Optimistic Concurrency Control where appropriate, while also enforcing an effective transaction isolation level as you would use with pessimistic concurrency control. For more information, see "Configuring CMP isolation level" in Chapter 4, "EJB CMP Tuning," in the *EAServer Performance and Tuning Guide*.

## Table-level timestamps

Beginning in version 4.1.3, you can configure table-level timestamp for CMP entity beans that use OCC. EAServer maintains the table-level timestamp as a single integer counter, incrementing the count for every update, insert, or delete in the main table. The conventional model for optimistic concurrency uses row-level timestamps. For more information, see "Concurrency control options" in Chapter 4, "EJB CMP Tuning," in the *EAServer Performance and Tuning Guide*.

## Soft locking

EAServer 4.2.2 introduces support for in-server soft locking of database rows used by EJB CMP entity beans that use the isolation level `repeatable_read_with_cache`. For more information, see "Concurrency control options" in Chapter 4, "EJB CMP Tuning," in the *EAServer Performance and Tuning Guide*.

## Generated entity bean subclasses

Beginning in EAServer 4.1.3, you can use the "Generated class" option for EJB 1.1 and 2.0 CMP entity beans. This option offers better performance than the "Automatic persistence" option, since the interaction between the storage component and the CMP implementation is more direct. If you import CMP entity beans from an EJB-JAR file, the "Generated Class" option is enabled by default. For existing CMP entity beans, you can configure it manually as described in "Generated entity bean subclasses" in Chapter 4, "EJB CMP Tuning," in the *EAServer Performance and Tuning Guide*.

# EJB session bean passivation

EAServer 4.2.2 introduces a new mechanism to support passivation of EJB stateful session beans based on the instance timeout property. Passivation is the process of removing an instance's data from memory and saving it to a database. Passivation allows the server to free the memory used by an instance while keeping the instance available for use by the client. When the client calls the component again, the instance is restored to memory. For more information, see "Using EJB activation and passivation" in Chapter 28, "Configuring Persistence for Stateful Session Components," in the *EAServer Programmer's Guide*.

# Message service enhancements

Beginning in EAServer 4.2.2, you can configure a default maximum size for message queues. The default.maximum property in the *MessageServiceConfig.props* file specifies the maximum queue size. For more information, see "Configuring the message service" in Chapter 8, "Setting up the Message Service," in the *EAServer System Administration Guide*.

# New features in version 4.2.1

EAServer 4.2.1 supports these new features:

* Web application form login enhancements

* EJB pass-by-reference support

* Database set-proxy support for EJB CMP entity beans

* SSL connection troubleshooting enhancement

## Web application form login enhancements

EAServer 4.2.1 adds support for direct form login, which allows you to access a Web application's protected content directly, without requiring the user to visit the Web application's form login page. For more information, see "Web application direct form login" in Chapter 3, "Using Web Application Security," in the *EAServer Security Administration and Programming Guide*.

## EJB pass-by-reference support

EAServer 4.2.1 adds support for the proprietary EJB pass-by-reference mechanism supported by some other J2EE vendors. For more information, see "Optimizing in-server EJB calls" in Chapter 3, "Component Tuning," in the *EAServer Performance and Tuning Guide*. This feature is not intended for new development, which should use standard EJB 2.0 local interfaces.

## Database set-proxy support for EJB CMP entity beans

EAServer 4.2.1 adds a new property for connection caches, to be set if the cache is used by an EJB CMP entity beans and the cache has set-proxy support enabled. The com.sybase.jaguar.conncache.ssa.systemid property specifies an alternate user name to set proxy to.

In an EJB CMP entity bean, the client user name is not available to set proxy to since the persistence engine runs as the system user. In this case, if a user name is specified with this property, the cache manager sets proxy to the specified user. Otherwise, it sets proxy to the cache's User Name.

## SSL connection troubleshooting enhancement

You can set the new security profile property
com.sybase.jaguar.security.logpeerIP. If this property is set to true in an SSL
listener's security profile, the client's IP address is logged when SSL
connections cannot be established. The default is false.

# New features in version 4.2

This section outlines the new features in EAServer 4.2.

## In-memory HTTP session replication

For Web applications that are deployed in an EAServer cluster, EAServer 4.2
can distribute HTTP session data using in-memory replication rather than
database storage. This feature can improve performance by avoiding the
overhead of writing to the database. For more information, see "Deploying
Web applications to a cluster" in Chapter 7, "Load Balancing, Failover, and
Component Availability," in the *EAServer System Administration Guide*.

## Pooled component destruction

EAServer 4.2 explicitly destroys pooled component instances before the server
shuts down. This allows you to perform cleanup operations in your component,
such as closing database connections. For more information, see "Pooled
component destruction" in Chapter 2, "Server Tuning," in the *EAServer
Performance and Tuning Guide*.

# New Features in EAServer Versions 4.1.3, 4.1.2, 4.1.1, and 4.1

# New features in version 4.1.3

EAServer 4.1.3 introduces the following:

- Performance tuning features

- HTTPS support for Web server plug-ins

- Jagtool merge_props command

## Performance tuning features

New EAServer 4.1.3 features for performance tuning include:

- Lightweight EJB container

- CMP JDBC wrapper drivers

- Thread monitors

- Entity instance caching with Oracle

- New server properties

- New connection cache properties

EAServer 4.1.3 also includes new component properties to improve the performance of EJB-CMP entity beans. These are presented together with the new 4.2.2 component properties described in "New EJB CMP component properties" on page 42.

## Lightweight EJB container

The performance of some inter-component calls in EAServer can be improved by enabling the lightweight container (LWC). The LWC co-exists with the standard container. The LWC provides optimized performance for EJB-to-EJB calls, while ensuring all expected EJB semantics are preserved such as pass by value when using remote interfaces. For more information, see "Optimizing in-server EJB calls" in in Chapter 3, "Component Tuning," in the *EAServer Performance and Tuning Guide*.

## CMP JDBC wrapper drivers

To improve performance this EAServer release includes customized JDBC drivers. You can use these wrapper drivers in connection caches that are used by CMP entity beans. The wrapper drivers offer better performance by allowing updates to be deferred to the end of each transaction and sent together as a command batch. Doing so improves performance by reducing network round trips between the database server and EAServer. The Sybase wrapper driver also supports automatic creation of semi-temporary stored procedures, which also improves performance. For more information, see "Using CMP JDBC wrapper drivers" in in Chapter 4, "EJB CMP Tuning," in the *EAServer Performance and Tuning Guide*.

## Thread monitors

EAServer 4.1.3 introduces a new Repository entity type, ThreadMonitor. Thread monitors provide a means to limit the execution time devoted to specified components and component methods. You can assign components and methods to a thread monitor to ensure that no more than a specified maximum number of threads will be active at any point executing the methods and components assigned to the monitor. You can also use thread monitors without a limit on the number of threads. Doing so allows you to use the monitor trace properties to record performance data. For more information, see "Thread monitors" in in Chapter 3, "Component Tuning," in the *EAServer Performance and Tuning Guide*.

### Entity instance caching with Oracle

You can use object caching when running EJB CMP entity beans. This feature works with any database. You can also configure change notification to notify EAServer's cache manager of changes to the underlying table rows. For more information, see see "Entity instance and query caching" in in Chapter 4, "EJB CMP Tuning," in the *EAServer Performance and Tuning Guide*.

### New server properties

Table 6-1 lists new server properties. Actual property names are prefixed with com.sybase.jaguar.server.

*Table 6-1: New server properties*

| Property | Description |
|---|---|
| com.sybase.jaguar.server.cmp_iso_level | Specifies a default transaction isolation level for component property com.sybase.jaguar.component.cmp_iso_level, which specifies the transaction isolation level for EJB CMP entity beans.. |
| com.sybase.jaguar.server.lwc | Enables the EJB lightweight container. |
| com.sybase.jaguar.server.lwc.debug | Enables debug output during lightweight container initialization. |
| com.sybase.jaguar.server.tx_retry | Sets the default value of the automatic transaction retry component property, com.sybase.jaguar.component.tx_retry. |

### New connection cache properties

EAServer 4.1.3 introduces com.sybase.jaguar.conncache.cmp_stats property to allow collection of statistics information when using an EAServer CMP JDBC wrapper driver—see "CMP JDBC wrapper drivers" on page 49.

## HTTPS support for Web server plug-ins

Beginning with EAServer version 4.1.3, you can establish HTTPS (HTTP over SSL) connections between your Web server redirector and EAServer, which provide an SSL communication link (data integrity and encryption services), but do not support authentication services using client-side certificates. For more information about establishing secure connections, see the *EAServer Security Administration and Programming Guide*. You should be familiar with SSL and Security Manager before establishing HTTPS connections. See Chapter 14, "Managing Keys and Certificates," in the *EAServer Security Administration and Programming Guide*.

See the *EAServer Installation Guide* for Web server redirector installation and configuration instructions.

## Jagtool merge_props command

Introduced in EAServer 4.1.3, the merge_props command merges or deletes property values for an entity. For more information, see the reference page in Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

## Java connection manager API change

The JCMCache.getCacheSize method has been replaced by the following methods:

```
public int getPoolSizeMax()
public int getPoolSizeMin()
```

JCMCache.getPoolSizeMax() returns the cache's maximum size. JCMCache.getPoolSizeMin() returns the minimum size. For more information, see the reference pages in Chapter 1, "Java Classes and Interfaces," in the *EAServer API Reference Manual*.

# New features in version 4.1.2

EAServer 4.1.2 introduces the following:

- JDK 1.4

- New performance tuning features

- JavaServer Pages compilation speed

- jagtool enhancements

- Client connection timeouts

- Web server redirector plug-in configuration

- Web application authentication

- APPLET context parameter for EJB clients

- Web application character set properties

- HTTP session attribute API change

# JDK 1.4

EAServer 4.1.2 adds support for running with JDK 1.4 as the in-server Java virtual machine.

# New performance tuning features

The following performance-related features are new in EAServer 4.1.2:

- Connection request pool size (Solaris only)

- Flow control

- Servlet buffer pool

- Servlet Java cache

- Direct-access pseudocomponents

## Connection request pool size (Solaris only)

On Solaris, EAServer 4.1.2 introduces a new listener property to configure the pool size for outstanding connection requests. When the server is very busy, all available threads may be in use when a connect request arrives. These pending connect requests are pooled until they can be handled. If the pool size is too small, client connection requests may time out before the server can handle the request. For more information, see "Listener tuning" in Chapter 2, "Server Tuning," in the *EAServer Performance and Tuning Guide*.

## Flow control

When the server is very busy with many client connections, client request threads may repeatedly conflict with each other for access to low-level system resources. Flow control provides a coarser level of granularity for synchronizing access to system resources by request threads. When enabled, flow control can improve performance by replacing multiple, serial choke points in the request processing sequence with a single choke point. Flow control can be enabled separately for HTTP and IIOP clients. For more information, see "Flow control" in Chapter 2, "Server Tuning," in the *EAServer Performance and Tuning Guide*.

## Servlet buffer pool

Internally, EAServer uses 4K and 8K temporary buffers when assembling servlet responses. EAServer 4.1.2 includes new server properties to tune the size of the 4K and 8K internal buffer pools. For more information, see "Servlet buffer pools" in Chapter 5, "Web Application Tuning," in the *EAServer Performance and Tuning Guide*.

## Servlet Java cache

This new EAServer 4.1.2 feature allows caching of servlet output in Java core memory, which offers a faster response than page caching for a limited class of servlets. For more information, see "Understanding HTTP response caching options" in Chapter 5, "Web Application Tuning," in the *EAServer Performance and Tuning Guide*.

## Direct-access pseudocomponents

In EAServer 4.1.2, you can generate special stubs and skeletons that improve the performance of pseudocomponent method calls. In a process called *marshalling*, regular CORBA stubs convert parameter and return values to the format required for IIOP network transport. Direct-access pseudocomponent stubs and skeletons improve performance by eliminating the marshalling step. For more information, see Chapter 34, "Creating and Using EAServer Pseudocomponents," in the *EAServer Programmer's Guide*.

# JavaServer Pages compilation speed

To reduce the compilation time for JavaServer Pages (JSPs), you can configure the class path for JSPs separately from the EAServer class path. Doing so can speed up the compilation of the JSPs. This can improve performance if your application requires runtime compilation, and can speed up the development cycle. For more information, see "Runtime compilation settings" in Chapter 5, "Web Application Tuning," in the *EAServer Performance and Tuning Guide*.

# jagtool enhancements

See Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide* for more information on using these tools. The following commands are new or enhanced in are new in EAServer 4.1.2:

| Command | Enhancement |
|---|---|
| gen_skels | New option to generate direct access skeletons:<br>`-pseudoskels true\|false` |
| gen_stubs | New option to generate direct access stubs:<br>`-pseudosstubs true\|false` |
| gen_stubsandskels | New option to generate direct access stubs and skeletons:<br>`-pseudostubsandskels true\|false` |
| gen_tlbreg | New. Generates type library (TLB) and registry (REG) files for ActiveX clients. This command is only available in Windows installations. |
| jmsflush | New. Flushes a message queue. |
| jmslist_listeners | New. Displays the names of the listeners attached to the specified message queue. |
| jmslist_messages | New. Lists the messages in the specified message queue. |

Beginning in EAServer 4.1.2, jagtool commands return status values that can be checked in scripts that call jagtool. The return status for each command is listed in the reference pages in Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

## Client connection timeouts

Response timeout properties have been added to the Java CORBA client ORB and EJB initial context. For applications that run in a cluster, these properties allow you to specify a time limit to receive a server response before the connection fails over to try another server in the cluster. Setting this property ensures that failover happens without an unreasonable delay. The property names are:

- com.sybase.CORBA.ConnectionTimeout for CORBA Java clients

- com.sybase.ejb.ConnectionTimeout for EJB clients

For more information, see these chapters in the *EAServer Programmer's Guide*:

- Chapter 8, "Creating Enterprise JavaBeans Clients"

- Chapter 12, "Creating CORBA Java Clients"

## Web server redirector plug-in configuration

For information on configuring the Web server redirector plug-ins, see the *EAServer Installation Guide* for your platform. Beginning in EAServer 4.1.2, these configuration options are now available for the Apache, iPlanet, and Netscape redirector plug-ins:

**Binding to a single server**   You can bind a redirector to a specific server so that if 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 *distributable*, you must bind the redirector to a specific server. If the redirector is bound to a server and the 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**   To trace requests from connectors, set this directive in the redirector configuration file:

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

When this directive is set, the connector appends the value of *ConnectorSessionId* 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 directive to the redirector configuration file:

```
Connector.SessionId ConnSID
```

EAServer writes this information to the HTTP request log:

```
10.22.85.66 - - [28/May/2002:12:11:09 -0800] "GET
/TestHTTPS/?ConnSID=2696_000000000000 HTTP/1.0" 200 51
10.22.85.66 - - [28/May/2002:12:11:24 -0800] "GET
/TestHTTPS/?ConnSID=2888_000000000000 HTTP/1.0" 304 0
10.22.85.66 - - [28/May/2002:12:11:40 -0800] "GET
/TestHTTPS/?ConnSID=2889_000000000000 HTTP/1.0" 304 0
10.22.85.66 - - [28/May/2002:12:11:40 -0800] "GET
/TestHTTPS/?ConnSID=2888_000000000001 HTTP/1.0" 304 0
10.22.85.66 - - [28/May/2002:12:11:40 -0800] "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 session ID is computed as:

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

## Web application authentication

"None" is no longer a Web application authentication option. The default authentication mechanism for a Web application is "HTTP Basic Authentication". Since Web Applications implement lazy authentication, if you want to authenticate your client, you must associate a role with a Web resource. The authentication options are:

- HTTP Basic

- Form authentication

- SSL mutual authentication

See Chapter 3, "Using Web Application Security," in the *EAServer Security Administration and Programming Guide* for more information.

## APPLET context parameter for EJB clients

Beginning in EAServer 4.1.2, EJB clients that run as applets can set the APPLET parameter for the javax.naming.InitialContext instance used to connect to EAServer. For more information, see Chapter 8, "Creating Enterprise JavaBeans Clients," in the *EAServer Programmer's Guide*.

## Web application character set properties

EAServer 4.1.2 introduces new Web application properties that enable you to specify input character sets. When the servlet engine initializes, it creates three mappings for each Web application, one for the input parameters, one for the input data, and one for JSP compilation as specified by these properties:

| Property name | Specifies character set for |
| --- | --- |
| com.sybase.jaguar.webapplication.charset.inputdata | Input data |
| com.sybase.jaguar.webapplication.charset.inputparam | Input parameters |
| com.sybase.jaguar.webapplication.charset.jspcompile | JSP compiler option |

For information on configuring these properties, see "Localizing Web applications" in Chapter 21, "Creating Web Applications," in the *EAServer Programmer's Guide*

## HTTP session attribute API change

If you call javax.servlet.http.HttpSession.setAttribute with null as the second parameter, you now get the same result as if you called javax.servlet.http.HttpSession.removeAttribute. In versions of EAServer earlier than 4.1.2, this caused an exception to be thrown.

# New features in version 4.1.1

EAServer 4.1.1 introduces these new features:

• Message Bridge for Java enhancements

• Java Secure Sockets Extension support

• Named instance pools

## Message Bridge for Java enhancements

EAServer 4.1.1 includes Message Bridge for Java™, version 2.1.1.
Enhancements in this version include:

- Improved support for XML Schema

- Performance improvements for schema import, code generation, and the
  graphical user interface.

- Additional samples.

For more information on these features, see the *Message Bridge for Java™
User's Guide*.

## Java Secure Sockets Extension support

EAServer 4.1.1 allows you to use Sun's Java Secure Sockets Extension (JSSE) classes
provide support for SSL connections. JSSE provides an alternative to the built-in SSL
implementations when secure connections are needed from an applet running in a Web
browser. To support JSSE, new properties have been added to the Java client runtime
classes. For information on using JSSE in Java clients, see Chapter 5, "Using SSL in
Java Clients," in the *EAServer Security Programming and Administration Guide*.

## Named instance pools

EAServer pools and reuses component instances to avoid the overhead of
creating and destroying instances. EAServer 4.1.1 provides named instance
pools to provide more administrative control over how components are pooled.
This feature allows you to control the number of component instances that
EAServer creates for a set of components. Instance pools provide a mechanism
to control the memory used to pool component instances. For example, you can
provide a larger instance pool to heavily used components. For more
information, see "Named instance pools" in Chapter 3, "Component Tuning,"
in the *EAServer Performance and Tuning Guide*.

# New features in version 4.1

EAServer 4.1 introduces these features:

- EJB 2.0 container-managed persistence enhancements

- Web services support

- Message Bridge for Java™

- Deployment enhancements

- Web server redirector for Microsoft Internet Information Server

- HTTP response caching enhancements

- Connection caching enhancements

- jagtool enhancements

- Servlet and Web application interface changes

- Maintaining authenticated sessions

- EJB 2.0 component behavior in EAServer 4.0 versus 4.1

## EJB 2.0 container-managed persistence enhancements

EAServer 4.1 adds support for ejbSelect methods and queries defined in EJB query language (EJB-QL), as required by the EJB 2.0 specification. ejbSelect methods allow you to run queries internal to your implementation. In your implementation, these methods must be declared private and abstract, and EAServer provides the runtime implementation. EJB-QL is the new standard language to define the database queries required to run finder methods and ejbSelect methods. At runtime, EAServer translates EJB-QL to SQL that is appropriate for the database type used by the component.

For more information on EAServer CMP support, see:

- "CMP enhancements" on page 69 describes CMP enhancements added to EAServer 4.0.

- Chapter 27, "Creating Entity Components," in the *EAServer Programmer's Guide* describes how to configure container managed persistence for EJB entity beans.

## Web services support

In EAServer 4.1, you can optionally install the Web Services Toolkit, which provides support for industry-standard Web services protocols such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Uniform Description, Discovery, and Integration (UDDI). SOAP provides a standard protocol for component invocations over HTTP connections, for example, you can expose a purchasing component to be called by business partners as a Web service. WSDL describes the service and can be used in client applications. You can also publish business and service information to a UDDI registry site on the Web and make your web service available to other users. SOAP provides a platform and language neutral means of accessing these services. With SOAP, WSDL, and UDDI, collaboration between business partners is made easier because interfaces between applications become standardized across disparate platforms.

The Web Services toolkit provides a Web application to run SOAP services, and graphical user interfaces to configure the SOAP services, create WSDL documents, and register services in UDDI registries.

For more information on these features, see the *Web Services Toolkit User's Guide*.

## Message Bridge for Java™

EAServer 4.1 includes Message Bridge for Java, a powerful, easy-to-use tool you can use to build applications that generate and consume documents and messages. Message Bridge generates Java classes that simplify the parsing and construction of XML documents. For more information, see the *Message Bridge for Java™ User's Guide*.

## Deployment enhancements

EAServer 4.1 has new features to simplify the deployment of applications to EAServer:

- When using Jaguar JAR format archives to deploy between EAServer installations, you can include merge files. A merge file allows you to modify or delete settings for entities that already exist. For example, you can create a merge file to install a package into the Jaguar server, without affecting the list of packages that are already installed.

- When using J2EE standard archive formats such as an EJB-JAR file, an application EAR file, or a Web application WAR file, you can define an EAServer supplemental deployment descriptor to configure properties that are required by EAServer but are not supported by the standard deployment descriptor format. EAServer creates a supplemental descriptor for any archive that you export from EAServer Manager.

For more information on these features, see Chapter 9, "Importing and Exporting Application Components," in the *EAServer System Administration Guide*.

## Web server redirector for Microsoft Internet Information Server

EAServer 4.1 adds support for integration with Web sites hosted by Microsoft Internet Information Server (IIS) on Windows platforms. You can host static content on IIS, and use the EAServer Redirector Plug-in for IIS to redirect servlet and JSP requests to EAServer. In addition to IIS, plug-ins for several other Web servers are available. For more information, see the *EAServer Installation Guide* for Windows.

## HTTP response caching enhancements

EAServer 4.1 adds support for caching of static content, and enhances the servlet response caching abilities introduced in EAServer 4.0.

Static page caching can increase performance by caching static files in memory. You can configure which files are cached and for how long on a per-server basis. For more information, see Chapter 5, "Web Application Tuning," in the *EAServer Performance and Tuning Guide*.

In EAServer 4.1, the entire response from JSPs that use forward or include commands is cached. Previously, you needed to configure caching separately for the forwarded or included page. In version 4.1, you can enable the Cache Entire Tree option for a Web component so that the response from all forwards and includes is cached together with content generated by the Web component itself. For more information on servlet response caching, see see Chapter 5, "Web Application Tuning," in the *EAServer Performance and Tuning Guide*.

## Connection caching enhancements

EAServer 4.1 adds connection cache support for Oracle OCI 9 connections to Oracle databases, and Sybase Open Client™ Client-Library™ connections to highly-available Adaptive Server Enterprise databases.

## jagtool enhancements

The administration automation tools, jagtool and jagant, have been enhanced with several new commands for EAServer 4.1, including:

• Several new commands to configure Java Message Service entities.

• The compilejsp command to compile JavaServer Pages.

• The configure command to execute XML configuration files.

• The exportconfig command to create XML configuration files.

For more information, see Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

## Servlet and Web application interface changes

The servlet filter interface required by EAServer 4.1 has changed from version 4.0. The setFilterConfig method is no longer supported (as of version 4.1); it has been replaced by the init and destroy methods.

For Web application lifecycle event listeners, the interfaces javax.servlet.ServletContextAttributeListener and javax.servlet.http.HttpSessionAttributeListener are both new for EAServer version 4.1. The corresponding interfaces from EAServer version 4.0, where "Attributes" was plural, are not supported in EAServer 4.1.

For more information on servlet filters and Web application lifecycle event listeners, see Chapter 23, "Using Filters and Event Listeners," in the *EAServer Programmer's Guide*.

## Maintaining authenticated sessions

EAServer 4.1 adds methods that allow a client to maintain an authenticated session across multiple transport sessions. In versions earlier than EAServer 4.1, if a client created a second component instance on the same server, or created a component instance on another server in the same cluster, the client was reauthenticated. A server can now authenticate and verify a client's credentials once, and, on subsequent authentication requests, the client provides a reference to the initial authentication.

To use the single authentication mechanism, you must use a custom authentication component that implements the CtsSecurity::AuthService interface.

CtsSecurity::SessionInfo::setName is a method that can be called only when your custom authentication component is running. When this method is called from the custom authentication component, the server sets the reference to the authenticated security credentials. When the client needs to be authenticated again, the custom authentication component returns the original principal name by calling CtsSecurity::AuthService::getCallerPrincipal(string alternate_name). For more information, see Chapter 10, "Creating and Using Custom Security Components," in the *Security Administration and Programming Guide*.

## EJB 2.0 component behavior in EAServer 4.0 versus 4.1

The effect of the com.sybase.jaguar.server.ejb.role.default property on EJB 2.0 components in EAServer 4.0 differs from EAServer version 4.1:

- The com.sybase.jaguar.server.ejb.role.default property is a server-wide property. To assign roles to all methods of an EJB 2.0 component in EAServer 4.0, you can assign individual roles to all methods of the component, or set the com.sybase.jaguar.server.ejb.role.default property to everybody.

- In EAServer 4.1, com.sybase.jaguar.server.ejb.role.default has been removed. To assign a role to any one method of an EJB 2.0 component, assign roles to all methods of the component. Otherwise, the server displays an error message when trying to execute the method that does not have a role assigned to it.

    If none of the methods of an EJB 2.0 component have roles assigned to them, authorization checks are not enforced and authorization is not performed. No error message is sent to the server log.

# New Features in EAServer 4.0

EAServer 4.0 includes these new features:

## J2EE 1.3 support

EAServer 4.0 adds features to support the Java Platform, Enterprise Edition specification, version 1.3. These features include:

- Enterprise JavaBeans 2.0 support, described in "EJB 2.0 support" on page 67.

- Java Message Service support, described in "Message service enhancements" on page 69.

- Enhanced Web application support to implement the Java Servlet 2.3 and JavaServer Pages (JSP) 1.2 specifications, described in "Servlet, JSP, and Web application enhancements" on page 70.

- Java API for XML Parsing (JAXP) integration, described in "Java XML parser integration" on page 72.

- The J2EE connector architecture, described in "J2EE connectors" on page 72.

- J2EE naming enhancements described in "J2EE naming enhancements" on page 72.

- J2EE 1.3 security features such as CSIv2 and the Java Authentication and Authorization Service (JAAS) API, described in "J2EE security enhancements" on page 73.

- J2EE 1.3 transaction features, such as transaction propagation and the Java Transaction Service (JTS), described in "Enhanced transaction support" on page 73.

# Performance enhancements

EAServer 4.0 includes the following features to improve application performance.

## JDK 1.3 and Hotspot VM integration

For improved performance, EAServer 4.0 can run with JDK 1.3 and the Java Hotspot virtual machine. On some platforms, EAServer can also use the Java Hotspot VM with JDK 1.2. The server JDK is determined by the command-line syntax that you use to start the server. For details, see Chapter 3, "Creating and Configuring Servers," in the *EAServer System Administration Guide*.

## Servlet response caching

EAServer 4.0 includes a servlet response cache to improve response times for Web clients. When page caching is enabled for a servlet or JSP, EAServer checks the cache before invoking the servlet. If an appropriate response is found in the cache, EAServer returns the contents of the cache, instead of calling the servlet. If a response is not found in the cache, EAServer invokes the servlet, then saves the response and the response headers in the cache.

Servlet response caching can be enabled and configured in EAServer Manager. No programming is required. For more information, see Chapter 5, "Web Application Tuning," in the *EAServer Performance and Tuning Guide*.

## Entity component instance and query caching

EAServer 4.0 supports instance and query caching for EJB entity beans and components of other types that use the entity design pattern. Caching can improve performance for entity components that use automatic persistence by minimizing the number of database select queries required for ejbLoad operations and finder method invocations. For more information, see Chapter 4, "EJB CMP Tuning," in the *EAServer Programmer's Guide*.

## In-memory state replication for stateful session components

EAServer 4.0 enhances the existing stateful failover architecture with support for failover in memory. This feature allows component instance state to be maintained on a pair of servers, without incurring the overhead of using a remote database to store component state. For more information, see Chapter 28, "Configuring Persistence for Stateful Session Components," in the *EAServer Programmer's Guide*.

# EJB 2.0 support

EJB 2.0 introduces support for message-driven beans, new home interface method syntax, local interfaces, and inter-vendor interoperability. EJB 2.0 also enhances the container managed persistence model defined in EJB 1.1.

## Message-driven beans

EJB 2.0 integrates the EJB component architecture with the Java Message Service (JMS) asynchronous messaging API. EJB 2.0 allows you to define message-driven bean components to respond to JMS messages. An MDB component is similar to an EJB stateless session bean, but the MDB component responds only to JMS messages and has no direct client interface.

For information on JMS and MDB components, see Chapter 31, "Using the Message Service," in the *EAServer Programmer's Guide*.

# Home interface methods

EJB 2.0 allows you to define business methods in the home interface for an entity bean and changes the syntax of create methods. For information on defining home interface methods, see Chapter 7, "Creating Enterprise JavaBeans Components," in the *EAServer Programmer's Guide*.

Create method syntax   In earlier versions, create methods were restricted to methods named create. In EJB 2.0, you can use any name that begins with create, such as createNewAccount.

Home interface business methods   You can add business methods to the home interface for an entity bean to perform operations that are not specific to a single instance. For example, a home business method might return the average employee salary. For each home business method, the entity bean's implementation class must have a method with the same name, except for the prefix ejbHome, and the same signature. For example, if the home interface declares:

```
public double averageSalary();
```

Then the implementation class must contain:

```
public double ejbHomeAverageSalary();
```

# Local interfaces

The EJB 2.0 architecture introduces local interfaces for calls to an EJB component from within the same Java virtual machine. In EAServer, you can use local interfaces for intercomponent calls, and for component invocations made from servlets hosted in the same server as the component.

Using local interfaces can improve performance, but in coding you must be aware that:

• Parameters are passed by reference rather than by copy, so object instances passed through a local invocation can be shared by the client and component. If the component modifies the object, the client sees the changes.

• Local interfaces are not location-transparent. The called component must be hosted in the same server process as the calling component.

## EJB 2.0 interoperability

EAServer 4.0 implements the interoperability requirements in the EJB 2.0 specification to allow interoperability with other EJB 2.0 servers. EAServer continues to support CORBA-2.2 based interoperability, for interacting with other CORBA-based application servers and to allow interoperability between EJB components hosted by EAServer and EAServer components of other types. For more information, see Chapter 9, "EAServer EJB Interoperability," in the *EAServer Programmer's Guide*.

## CMP enhancements

EJB 2.0 enhances the Container-Managed Persistence (CMP) model for entity beans as follows:

- The deployment descriptor more fully describes the persistent fields in the bean and the required database queries, making for less work after deploying an EJB-JAR file that contains CMP entity beans.

- CMP entity beans in the same EJB-JAR (which maps to an EAServer package) can have container-managed relationships. For example, an *Order* bean may have an *items* field that consists of a collection of *Inventory* bean instances representing the items being purchased. Or, an *Employee* bean may be related to itself, with *manager* and *employees* fields that contain Employee instances.

For more information on EAServer CMP support, see Chapter 27, "Creating Entity Components," in the *EAServer Programmer's Guide*.

# Message service enhancements

EAServer 4.0 provides an implementation of the Java Message Service (JMS) API, message-driven bean support, and easier management of the message service.

## Java Message Service API support

EAServer 4.0 provides implementations of the JMS API drivers, layered on top of the existing message service API. The JMS specification defines two messaging models: point-to-point for sending messages to specific message consumers, and Publish/Subscribe for publishing messages that are available to all consumers. For more information, see Chapter 31, "Using the Message Service," in the *EAServer Programmer's Guide*.

## Message-driven bean support

Message driven bean (MDB) components integrate the EJB 2.0 and Java Message Service architectures. See "Message-driven beans" on page 67 for more information.

## EAServer Manager enhancements

You can now use EAServer Manager to add and configure the message service components that allow you to send and receive messages. For more information, see Chapter 8, "Setting up the Message Service," in the *EAServer System Administration Guide*.

# Servlet, JSP, and Web application enhancements

EAServer 4.0 supports the Servlet 2.3 and JSP 1.2 specifications, and adds support to connect Web applications hosted in EAServer to popular Web servers such as Apache and Netscape.

## Servlet filters and application lifecycle event handlers

EAServer 4.0 supports servlet filters. Using servlet filters, you can preprocess servlet requests and postprocess servlet responses. You can install code that executes in response to changes in the servlet context or HTTP sessions associated with your Web application. For details on these features, see Chapter 23, "Using Filters and Event Listeners," in the *EAServer Programmer's Guide*.

## JSP 1.2 support

EAServer 4.0 supports the JSP 1.2 specification. JSP 1.2 extends JSP 1.1 in a number of ways:

- Uses Servlet 2.3 as the foundation for its semantics.

- Defines the XML syntax for JSP pages.

- Provides for translation-time validation of JSP pages. A new compilation phase has been added that gives custom tag libraries the opportunity to examine an XML view of the parsed page, and throw a translation time exception if problems are detected.

- The ability for tag libraries to include event listener classes. The listeners are listed in the tag library descriptor and the JSP container automatically instantiates the listener classes and registers them in a similar way to *web.xml*. Essentially, the mechanism locates the type-library descriptors (TLDs) in the Web Application (either in *WEB-INF/classes* or *WEB-INF/lib*), reads their `<listener>` elements, and regards them as an extension of those listed in *web.xml*.

- Better specification of the tag handler contract – tag handlers are Java classes that implement specific servlet interfaces. JSP 1.2 introduces some new interfaces and changes some existing interfaces to provide better support.

- Improvements on authoring support – extends tag library functionality.

- Better I18N support – implements the `javax.servlet.ServletResponse.setContentType()` method to provide support for dynamic content-type.

- Fixes the "flush before you include" limitation in JSP 1.1.

For information on creating JSPs in EAServer, see Chapter 24, "Creating JavaServer Pages," in the *EAServer Programmer's Guide*.

## Standard extensions

In accord with the Servlet 2.3 specification, EAServer allows you to install extensions and define the extensions required by a Web application. If you import a Web application (in WAR format) that requires extensions that are not installed, EAServer Manager warns you of the unfulfilled dependencies. For more information, see Chapter 21, "Creating Web Applications," in the *EAServer Programmer's Guide*

## Web server redirector plug-in

The Web server redirector plug-in allows communication between the Web server and the EAServer HTTP protocol listener. You can integrate a Web server and EAServer by installing the redirector plug-in on the Web server host. Client requests are sent directly to the Web server, which forwards requests to EAServer that need to access EAServer resources. EAServer processes the requests and returns the results back to the Web server. For details, see the *EAServer Installation Guide*.

# Java XML parser integration

EAServer 4.0 includes support for JAXP (Java API for XML Parsing) 1.1. The package includes the industry-standard DOM and SAX APIs, Crimson SAX and DOM parsers, and the Xalan XSLT transformer from Apache. You can configure the parser and transformer implementations for servers, components, Web applications, and Application clients. This implementation fulfills the requirements of JAXP 1.1 stated in Section 6.11 of the J2EE 1.3 specification. For more information, see Chapter 36, "Configuring Java XML Parser Support," in the *EAServer Programmer's Guide*.

# J2EE connectors

The J2EE connector architecture defines a resource adapter, which acts as a connection factory and can provide a Java client with connections to multiple enterprise information systems. Connectors are typically used to allow EJB components, servlets, and JSPs to access remote databases. For more information, see Chapter 4, "Database Access," in the *EAServer System Administration Guide*.

# J2EE naming enhancements

In accordance with J2EE 1.3 and EJB 2.0 requirements, the EAServer naming service now supports logical names for:

- Message-driven beans

- JMS resource connection factory references

- Resource environment references

- J2EE connectors

- EJB components hosted on other EJB 2.0 application servers, using interoperable naming URLs

For more information, see Chapter 5, "Naming Services," in the *EAServer System Administration Guide*.

# J2EE security enhancements

EAServer 4.0 implements J2EE version 1.3 security requirements including the JAAS (Java Authentication and Authorization Services) API and CORBA Secure Interoperable version 2 protocol (CSIv2). CSIv2 allows security context propagation across servers to satisfy EJB 2.0 interoperability requirements. JAAS provides a framework and standard programming interface for authenticating users and assigning privileges.

For more information, see the following chapters in the *EAServer Security Administration and Programming Guide*:

- Chapter 2, "Securing Component Access,"

- Chapter 11, "Using the JAAS API,"

# Enhanced transaction support

EAServer 4.0 supports the Java Transaction Service Transaction Manager. This enables EAServer to control the scope and duration of transactions across multiple resource managers. EAServer also supports transaction interoperability, which allows transactional components on multiple servers to participate in the same transaction. For more information, see Chapter 2, "Understanding Transactions and Component Lifecycles," in the *EAServer Programmer's Guide*.

# HTTP 1.1 support

EAServer 4.0 offers support for HTTP/1.1. EAServer 4.0 supports features listed as must and required for *origin server* and *client* in the W3C spec for HTTP/1.1. For more information about HTTP/1.1, see the HTTP/1.1 specification at http://www.w3.org/Protocols/rfc2616/rfc2616.html.

# Enhanced C++ component support

Beginning in EAServer 4.0, you can run C++ components in a separate process from the host server. This allows you to run legacy C and C++ code as a Jaguar component, while eliminating the risk incurred by running C++ code in the same process as the host server. EAServer 4.0 also introduces support for C++ components in mixed-platform clusters. For more information, see Chapter 14, "Creating CORBA C++ Components," in the *EAServer Programmer's Guide*.

# Java deployment enhancements

EAServer 4.0 provides greater control over application class loading and some Java-to-IDL and IDL-to-Java code generation optimizations.

## New custom class loaders

EAServer 4.0 extends the custom class loader model with package, application, and server class loaders. This enhancement allows components and Web applications to share instances of common classes, while still supporting hot refresh of application classes. For more information, see Chapter 30, "Configuring Custom Java Class Lists," in the *EAServer Programmer's Guide*.

## Deployment and code generation performance

EAServer 4.0 introduces Java-to-IDL and IDL-to-Java code generation optimizations. These optimizations can increase your productivity when you deploy Java based applications with complicated component class hierarchies and when you generate Java stubs for these applications.

The J2EE EAR importer, EJB JAR importer, and the Java code generator have been enhanced to allow you to specify a code generation strategy as described in the following table:

| Generation strategy | Specifies |
| --- | --- |
| Full | Always generates Java or IDL types for every interface and type. |
| Incremental | Compares IDL and Java timestamps to determine whether new versions must be generated. Java timestamps are maintained in a database in the EAServer repository. |
| Optimistic | Similar to Incremental, except that the check for changed types in each IDL module or Java package ends if the first comparison indicates no change. |

For information on using these options when importing EAR or JAR files or when generating Java stubs, see the online help in EAServer Manager.

# Systems management enhancements

EAServer 4.0 provides automated tools for deployment, configuration, and installation, simplifications to the server and tool scripts, and consolidation of the run-time Java classes.

## Server and EAServer Manager start-up scripts

The scripts to start EAServer Manager and the server have been simplified. For example, one script starts servers for all JDK versions. For more information, see the *EAServer System Administration Guide*.

# Globally available connection caches

Connection caches are now available to all servers run from the same EAServer installation. You no longer need to install connection caches into a server to allow access from components on that server.

# Java class file consolidation

EAServer's runtime Java classes have been consolidated into these JAR files:

- *$JAGUAR/java/lib/easserver.jar*, required in the CLASSPATH for server-side Java code.

- *$JAGUAR/java/lib/easclient.jar*, required in the CLASSPATH for server-side and client-side Java code.

- *$JAGUAR/java/lib/easj2ee.jar*, required in the CLASSPATH for server-side and client-side Java code.

# Automated installation and configuration tools

EAServer 4.0 includes tools to automate deployment, configuration, and installation. These features are useful if you are embedding EAServer in your own software installation.

For automated deployment and configuration, use these tools:

- jagtool is a command-line interface that allows you to automate some of EAServer's development and deployment tasks.

- jagant lets you run jagtool commands from Jakarta Ant build files.

For more information, see Chapter 12, "Using jagtool and jagant," in the *EAServer System Administration Guide*.

EAServer 4.0 also includes a script-driven installation utility that you can use to automate EAServer software installation. For more information, see the the *EAServer Installation Guide*.

# Repository versioning

Repository versioning allows you to save implementation and configuration information for your EAServer applications. For more information on this feature, see Chapter 10, "Using Repository Versioning," in the *EAServer System Administration Guide*.