

What's New in EAServer

Document ID: DC38032-01-0600-01

Last revised: July 2006

Topic	Page
Lightweight server architecture	2
Jaguar server and the JAGUAR environment variable	2
Web-based user interface	2
Setting up the Management Console	3
Ant-based configuration	4
Customized deployment	4
Migration tool	5
Custom class loaders	5
Improved monitoring	6
Connection Management	6
Messaging features	7
JNI compiler	8
Systems Management	9
Caching features	9
Client features	9
Clustering features	10
Unsupported features	10

Copyright 1997-2006 by Sybase, Inc. All rights reserved. Sybase, SYBASE (logo), ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Advantage Database Server, Afaria, Answers Anywhere, Applied Meta, Applied Metaconnecting, App-Modeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Translator, ASEP, Avaki, Avaki (Arrow Design), Avaki Data Grid, AvantGo, Backup Server, BayCam, Beyond Connected, Bit-Wise, BizTracker, Certified PowerBuilder Developer, Certified SYBASE Professional, Certified SYBASE Professional Logo, ClearConnect, Client-Library, Client Services, CodeBank, Column Design, ComponentPack, Connection Manager, Conveyor/DM, Copernicus, CSP, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DataWindow .NET, DB-Library, dbQueue, Dejima, Dejima Direct, Developers Workbench, DirectConnect Anywhere, DirectConnect, Distribution Director, Dynamic Mobility Model, e-ADK, E-Anywhere, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, EII Plus, Electronic Case Management, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Client/Server, Enterprise Manager, Enterprise Portal (logo), Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, eremote, Everything Works Better When Everything Works Together, EWA, ExtendedAssist, Extended Systems, ExtendedView, Financial Fusion, Financial Fusion (and design), Financial Fusion Server, Formula One, Fusion Powered e-Finance, Fusion Powered Financial Destinations, Fusion Powered STP, Gateway Manager, GeoPoint, GlobalFIX, iAnywhere, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InstaHelp, Intelligent Self-Care, InternetBuilder, iremote, irLite, iScript, Jaguar CTS, JConnect for JDBC, KnowledgeBase, Legion, Logical Memory Manager, MDM Anywhere, Mach Desktop, Mail Anywhere Studio, Mainframe Connect, Maintenance Express, Manage Anywhere Studio, MAP, M-Business Anywhere, M-Business Channel, M-Business Network, M-Business Suite, MDI Access Server, MDI Database Gateway, media.splash, Message Anywhere Server, MetaWorks, MethodSet, mFolio, Mirror Activator, ML Query, MobiCATS, MobileQ, MySupport, Net-Gateway, Net-Library, New Era of Networks, Next Generation Learning, Next Generation Learning Studio, O DEVICE, OASIS, OASIS logo, ObjectConnect, ObjectCycle, OmniConnect, OmniQ, OmniSQL Access Module, OmniSQL Toolkit, OneBridge, Open Biz, Open Business Interchange, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Partnerships that Work, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, Pharma Anywhere, PhysicalArchitect, Pocket PowerBuilder, PocketBuilder, Power++, Power Through Knowledge, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, Powering the New Economy, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Pylon, Pylon Anywhere, Pylon Application Server, Pylon Conduit, Pylon PIM Server, Pylon Pro, QAnywhere, Rapport, Relational Beans, RemoteWare, RepConnector, Report Workbench, Report-Execute, Replication Agent, Replication Driver, Replication Server Manager, Replication Toolkit, Resource Manager, RFID Anywhere, RW-Display-Lib, RW-Library, SAFE, SAFE/PRO, Sales Anywhere, Search Anywhere, SDF, Search Anywhere, Secure SQL Server, Secure SQL Toolset, Security Guardian, SharePool, ShareLink, SKILLS, smart.partners, smart.parts, smart.script, SOA Anywhere Trademark, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFI, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQL Stage III Engineering, Startup.Com, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Development Framework, Sybase Financial Server, Sybase Gateways, Sybase IQ, Sybase Learning Connection, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase Synergy Program, Sybase Virtual Server Architecture, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SybFlex, SybMD, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, The Enterprise Client/Server Company, The Extensible Software Platform, The Future Is Wide Open, The Learning Connection, The Model For Client/Server Solutions, The Online Information Center, The Power of One, TotalFix, TradeForce, Transact-SQL, Translation Toolkit, Turning Imagination Into Reality, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for Unicode, Viafone, Viewer, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server, XcelleNet, XP Server, XTNDAccess and XTNDConnect are trademarks of Sybase, Inc. or its subsidiaries. 05/06

Lightweight server architecture

A lightweight J2EE container based on a dynamic Java-components framework provides a modular, high-performance container with a small footprint for executing standard J2EE components. The container permits the execution of J2EE 1.4 applications, including Enterprise JavaBean (EJB) and Web components and the JMS API, on anything from high-end servers to mobile devices, such as a handheld or tablet PC running Windows XP. The core packages are 100% Java and portable to all JDK 1.4+ platforms. Native code is used only for optional modules; for example, operating system login modules, and for supporting CORBA and PowerBuilder™ components.

Jaguar server and the JAGUAR environment variable

In earlier versions of EAServer, the name of the preconfigured server was Jaguar, and the environment variable that identified the EAServer installation directory was JAGUAR. In EAServer 6.0, the preconfigured server is assigned the name of the machine on which EAServer is installed, and the name of the environment variable that identifies the EAServer installation directory is DJC_HOME. DJC reflects the server's new architecture, which uses dynamic Java components.

Web-based user interface

In EAServer 6.0, the Management Console, a Web-based user interface, replaces EAServer Manager as the graphical user interface to EAServer. The Management Console provides functionality similar to EAServer Manager, including support for servlet and JSP-based Web applications. The Management Console uses JavaScript, but is fully functional if JavaScript is disabled.

The Management Console provides graphical administration facilities for EAServer, including support for development, deployment, and runtime monitoring of applications. The Management Console enables users to perform administrative tasks from a Web browser, including managing files, monitoring system components, and editing text files.

Setting up the Management Console

To use the Management Console, you must be the `admin@system` user or belong to the `admin-role` role. For backward compatibility, `jadadmin` is treated as an alias for `admin@system`. When you install EAServer, the installer prompts you to enter the password. After you install EAServer, you can reset the administrative password using the `set-admin-password` script.

❖ **Setting the password for the administrative user**

- 1 At a command prompt, change to the EAServer *bin* directory.
- 2 On Solaris, run:

```
set-admin-password.sh
```

On Windows, run:

```
set-admin-password.bat
```

The system prompts you to enter a password, which must contain at least six characters, and one of those characters must be a digit. To configure these password requirements, see Chapter 10, “Security Configuration Tasks,” of the *EAServer Security Administration and Programming Guide*.

❖ **Connecting to EAServer using the Management Console**

Start the server and log in to the Management Console.

- 1 In a Web browser, access the following URL, where *host* is the name of the machine where EAServer is running; 8000 is the default HTTP port number.

```
http://host:8000/console
```

- 2 In the login window, enter:

- Username – `admin`.
- Password – the password that you set for this user.

The Management Console logs in to the local server.

The security domain “system” is the default domain. If you create a user in another domain, you must specify the domain when you log in; for example, `userName@domain`. The Management Console and the `wlogin` command line tool default to the system security domain.

When you log in to the Management Console, you can use the predefined connection profile “Local Domain,” or you can define a new connection profile.

See Chapter 2, “Management Console Overview,” in the *EAServer System Administration Guide*.

Ant-based configuration

In EAServer 6.0, Ant configuration scripts are used to define EAServer modules. The installation includes default configurations for several module types, including data sources, database types, EJB providers, export configurations, and thread monitors. Using the Management Console, you can view the predefined Ant scripts by expanding the Configuration Files folder, and selecting the file; the contents of the file display in the right pane.

If you use the Management Console to deploy a module, an Ant configuration script that defines the entity properties and component attributes is automatically generated.

See the *EAServer Automated Configuration Guide*.

Customized deployment

You can create customized deployment modules to export parts of a server for deployment, synchronization, or backup. These modules, called export configurations, include sets of EAServer files, Java classes, and entities (application clients, applications, connectors, EJBs, and Web applications). Each export configuration may include information about cluster synchronization and dependencies to other modules.

You can use an export configuration to:

- Export an entity, such as a Web application
- Create a backup of a server or parts of a server
- Define a set of files to be synchronized to another server in a cluster

- Create specific deployment options for a server; for example, you can create a server without a Web container, or a server that contains only an EJB container

See Chapter 7, “Exporting Server Modules,” in the *EAServer System Administration Guide*.

Migration tool

The migrate command line tool enables you to migrate J2EE applications, Web applications, Web services, EJBs, connectors, application clients, roles, configured queues, configured topics, queue connection factories, topic connection factories, and other entities from EAServer 5.x to EAServer 6.0. The migrate tool includes a -gui option, which allows you to use a graphical user interface to migrate EAServer.

For complete details, see the *EAServer Migration Guide*.

Custom class loaders

EAServer uses custom class loaders to allow deploying and updating components and applications without restarting the application server.

In Java, a *class loader* loads the Java classes used by an application. Most applications use the Java system class loader, which loads classes from the directories and JAR files specified by the CLASSPATH environment variable. In the normal Java program configuration, you must restart a program or server to begin using updated Java classes. EAServer uses custom class loaders to allow *hot refresh* of Web application classes and Java components without restarting the server.

See Chapter 10, “Configuring Java Class Loaders,” in the *EAServer System Administration Guide*.

Improved monitoring

New monitoring features include:

- Browse name bindings – displays all registered name bindings and their binding scope (client, cluster, server, package, or component).
- Client and server tracing options – RMI-IIOP, JMS, and SQL tracing.
- Tracing and profiling for EJB and Web components – you can enable tracing in Ant configuration scripts or on the command line. You can enable profiling in Ant configuration scripts, and use the Management Console or Web queries to view statistics in spreadsheets.
- Statistics view in spreadsheet format – you can display statistics in a spreadsheet format by accessing the statistics URL from either a Web browser or a spreadsheet program, such as Microsoft Excel.
- Runtime monitoring – accessed using the Statistics view in the Management Console.
- Improved exception logging and exception cross referencing – application and system exceptions can be enabled at the server level.
- Transaction cross referencing – logs information about SQL statements and JMS operations that are executed within a transaction.
- Thread monitors – available for CORBA, PowerBuilder, Web, and EJB components. Named monitors can be associated with components in Ant configuration scripts. Active and waiting threads visible in the Sybase Management Console.

See Chapter 11, “Runtime Monitoring,” in the *EAServer System Administration Guide*.

Connection Management

In EAServer 6.0, each connection manager that supports C/C++ and PowerBuilder components has its own connection library. In earlier versions of EAServer, a single connection library *libjcm.dll* (Windows) or *libjcm.so* (Solaris) handled all connection management for native connections. Separating the connection managers allows for smaller client or server footprints, and simplifies connection management.

C/C++ and PowerBuilder components require connections that are created using C/C++ code. To support C/C++ connections, EAServer 6.0 provides the new database types listed in Table 1. To manage each C/C++ connection, EAServer creates a Java object, which supports the JDBC API.

Table 1: Connection libraries

Database type	Java connection library	C/C++ connection library
JCM_Sybase	<i>libjcm_sybase.dll</i>	<i>libct.dll</i>
JCM_Odbc	<i>libjcm_odbc.dll</i>	<i>odbcad32.dll</i>
JCM_Odbc_Unicode	<i>libjcm_odbcu.dll</i>	<i>odbcad32.dll</i>
JCM_Oracle	<i>libjcm_oracle.dll</i>	<i>oci.dll</i>
JCM_Oracle_Unicode	<i>libjcm_oracleu.dll</i>	<i>oci.dll</i>

See the *EAServer CORBA Components Guide*.

Messaging features

New messaging features include:

- Support for the complete JMS 1.1 API.
- Message-driven beans (MDBs) can receive message types other than JMS. A J2EE Connector Architecture (JCA) 1.5 resource adapter integrates messaging providers with the container.
- Messages can be processed in the order in which they are received, even when multiple consumers access the same queue.
- Duplicate messages can be ignored.
- Application servers running on a mobile device can initiate pull and push operations; both online and offline modes are supported.
- Header and system properties can be used to construct source or target queue addresses; for pull, only system properties can be used.
- You can log in to the JMS provider using IIOP single sign-on scripts.
- For JMS clients, a new initial context factory class `com.sybase.jms.client.InitialContextFactory` replaces `com.sybase.jms.InitialContextFactory`.
- The `com.sybase.jms.client.InitialContextFactory` class performs JNDI lookups to resolve names that are not preconfigured.

- The `com.sybase.jms.client.InitialContextFactory` class supports four provider URL forms:
 - `iiop://host:port` – uses IIOP to access the message service.
 - `jms-provider:jmsProviderName` – uses the named JMS provider.
 - `run-server.server_name` – uses IIOP to access the message service, starts the server in-process if it is not already running, and selects the server’s first IIOP listener as the provider URL.
 - `start-server.server_name` – uses IIOP to access the message service, starts the server out-of-process if it is not already running, and selects the server’s first IIOP listener as the provider URL.

See the *EAServer Java Message Service User’s Guide*.

JNI compiler

The Java Native Interface (JNI) compiler is a command line tool that can generate:

- C++ proxies for accessing Java classes and interfaces, which are more productive and less error prone than direct JNI calls.
- C++ classes for implementing Java native methods, which are more efficient than IDL-based component calls from Java to C++ or vice versa.

You may want to use the JNI compiler:

- For developing C++ clients that talk to EJB components or JMS using the new EJB/JMS client runtime features (for example, data compression), which are not available in our CORBA C++ client runtime.
- When you need to call legacy C/C++ code from your components, and you would prefer to write EJBs with a few native methods than to write CORBA C++ components.

See Chapter 13, “JNI Compiler,” in the *EAServer System Administration Guide*.

Systems Management

Systems Management uses the MX4J implementation of the Java Management Extension (JMX) version 3.0.1 API. Systems Management exposes management and monitoring functionalities to other applications, such as the Simple Network Management Protocol (SNMP) and JMX-compliant management applications. You can also enable Systems Management to support other management applications, which use different protocols, by adding connectors and protocol adaptors. See Chapter 14, “Systems Management,” in the *EAServer System Administration Guide*.

Caching features

New caching features include:

- Partitioned entity cache – an active data partition can be cached and queries run in memory, while preserving consistency using commit-time validation.
- Entity query batching – optimized stored procedures reduce the number of queries from an application server to a database.
- Result caching – Session bean and CORBA component methods can be enabled for caching, and a database trigger can be used (instead of timeouts) to ensure transactional consistency of cached results.

Client features

New client features include:

- Automatic server startup – if the application server is not running, EJB and JMS clients can use special provider URLs to start the server automatically.
- Dynamic stub generation – stubs are generated automatically and compiled when first used; regenerated and recompiled if a component’s remote interface changes.
- IIOP data compression – available to EJB and CORBA Java clients; can use the `InitialContext` property.

Clustering features

New clustering features include:

- Cluster partitions – you can define partitions with primary and back-up servers, then configure individual components to be serviced only by the servers in a specific partition. Can replace “hot standby.”
- Automatic failover/request and response logging – to ignore duplicate calls during failover, session bean methods that return void can be enabled for request logging, and non-void session bean methods can be enabled for response logging.

Unsupported features

This section describes features that were supported in earlier versions of EAServer but are not supported in version 6.0.

- **ActiveX** ActiveX components are no longer supported. ActiveX CORBA clients running with EAServer 5.x can access EAServer 6.0. Use EAServer 5.x to generate *.tlb* and *.reg* files to use with your ActiveX-enabled IDE.

.NET clients will be supported in a future EAServer version.

- **Component creation support** Unlike EAServer Manager in earlier versions of EAServer, the Management Console does not support manual creation of standard J2EE components such as EJBs or Web applications. Instead, use industry standard development tools such as PowerBuilder, NetBeans, IntelliJ, Eclipse, and JBuilder to package these components in J2EE archives, then deploy them to EAServer with the Management Console, the command-line tools, or Ant.
- **CORBA clients** The initial context factory implementation class `com.sybase.jaguar.jndi.CosNaming.JaguarInitialContextFactory` is no longer supported.
- **J2EE module property sheets** Property sheets for J2EE modules are no longer supported. You can use the tools described for component creation support to edit J2EE deployment descriptors.
- **Management API** The `com.sybase.jaguar.management.*` API package is no longer supported.

- **MASP** The EAServer Methods As Stored Procedures (MASP) interface is not supported in EAServer 6.0.
MASP will be supported in a future EAServer version.
- **OCI 7 and 8** EAServer 6.0 does not support OCI 7 or OCI 8 native connection caches.
- **OTS and DTC transaction models** The transaction manager no longer supports either the OTS or DTC transaction models.
- **Static page caching** In earlier versions of EAServer, static page caching was used to improve JNI access from a C-language Web server to servlets and JSPs. All EAServer 6.0 code is Java-based, so you need no longer optimize JNI calls.
- **TDS.idl** *TDS.idl* is no longer included in EAServer.

