SYBASE®

Migration Guide

EAServer

6.0

DOCUMENT ID: DC00485-01-0600-01

LAST REVISED: July 2006

Copyright © 1997-2006 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

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 Metacomputing, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Translator, APT-Library, 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, Convoy/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 Connect, Enterprise Data Studio, 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, M2M 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, Replication Server Manager, Replication Toolkit, Resource Manager, RFID Anywhere, RW-DisplayLib, RW-Library, SAFE, SAFE/PRO, Sales Anywhere, Search Anywhere, SDF, Search Anywhere, Secure SQL Server, Secure SQL Toolset, Security Guardian, ShareSpool, ShareLink, SKILS, 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/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, 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

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book

Subject

This book contains information about migrating EAServer 5.*x* resources and entities to an EAServer 6.0 installation. You cannot upgrade an EAServer 5.*x* installation to EAServer 6.0 directly, instead you must install EAServer 6.0 then follow the instructions in this book to migrate your EAServer 5.*x* resources.

Audience

This book is for anyone responsible for installing and configuring the EAServer runtime environment.

How to use this book

Chapter 1, "Before Migrating to EAServer 6.0," contains information you need to know before you migrate.

Chapter 2, "Migrating to EAServer 6.0," contains instructions for migrating your EAServer 5.*x* entities to EAServer 6.0.

Chapter 3, "Post-migration tasks," contains information you need to know after you complete the migration.

Chapter 4, "EAServer 6.0 Properties," describes the EAServer 5.*x* properties and the 6.0 equivalents.

Related documents

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

What's New in EAServer 6.0 summarizes new functionality in this version.

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

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

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

The *EAServer CORBA Components Guide* explains how to:

- Create, deploy, and configure CORBA and PowerBuilder™ components and component-based applications

- Use the industry-standard CORBA and Java APIs supported by EAServer

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

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

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

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

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

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

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

The *EAServer System Administration Guide* explains how to:

- Start the preconfigured server and manage it with the Sybase Management Console
- Create, configure, and start new application servers
- Define database types and data sources
- Create clusters of application servers to host load-balanced and highly available components and Web applications
- Monitor servers and application components

- Automate administration and monitoring tasks with command line tools

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

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

- Support for standard Web services protocols such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Uniform Description, Discovery, and Integration (UDDI)
- Administration tools for deployment and creation of new Web services, WSDL document creation, UDDI registration, and SOAP management

The *EAServer Troubleshooting Guide* describes procedures for troubleshooting problems that EAServer users may encounter. This document is available only online; see the EAServer Troubleshooting Guide at http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.eas_5.2.eastg/html/eastg/title.htm.

jConnect for JDBC documents EAServer includes the jConnect™ for JDBC™ 6.0.5 driver to allow JDBC access to Sybase database servers and gateways. The *jConnect for JDBC 6.0.5 Programmer's Reference* is available on the Sybase Product Manuals Web site at http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.jconnjdbc_6.05.prjdbc/html/prjdbc/title.htm&toc=/com.sybase.help.jconnjdbc_6.05/toc.xml.

Sybase Software Asset Management User's Guide EAServer includes the Sybase Software Asset Management license manager for managing and tracking your Sybase software license deployments. The *Sybase Software Asset Management User's Guide* is available on the Getting Started CD and in the EAServer 6.0 collection on the Sybase Product Manuals Web site at http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.eas_6.0/title.htm.

Conventions

The formatting conventions used in this manual are:

Formatting example	To indicate
commands and methods	When used in descriptive text, this font indicates keywords such as: • Command names used in descriptive text • C++ and Java method or class names used in descriptive text • Java package names used in descriptive text • Property names in the raw format, as when using jagtool to configure applications rather than the Web Management Console

Formatting example	To indicate
variable, *package*, or *component*	Italic font indicates: • Program variables, such as *myCounter* • Parts of input text that must be substituted, for example: *Server*.log • File names • Names of components, EAServer packages, and other entities that are registered in the EAServer naming service
File | Save	Menu names and menu items are displayed in plain text. The vertical bar shows you how to navigate menu selections. For example, File | Save indicates "select Save from the File menu."
`package 1`	Monospace font indicates: • Information that you enter in the Web Management Console, a command line, or as program text • Example program fragments • Example output fragments

Other sources of information

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

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

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

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

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

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

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

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

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Select Products from the navigation bar on the left.
3. Select a product name from the product list and click Go.
4. Select the Certification Report filter, specify a time frame, and click Go.
5. Click a Certification Report title to display the report.

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

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

1. Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
2. Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

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

1. Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.
2. Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
3. Select a product.
4. Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

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

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

Accessibility features

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

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

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

1 Start Eclipse.

2 Select Help | Help Contents.

3 Enter `Accessibility` in the Search dialog box.

4 Select Accessible User Interfaces or Accessibility Features for Eclipse.

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

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

If you need help

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

CHAPTER 1

Before Migrating to EAServer 6.0

Introduction

This book provides information about how to migrate various components and entities from an EAServer 5.*x* installation to EAServer 6.0.

For J2EE entities in EAServer 6.0, do not configure repository properties directly. Properties are derived either from J2EE deployment descriptor properties or from a user configuration file.

Non-EJB components and entities that do not correspond to J2EE entities use repository properties in EAServer 6.0. CORBA, C++, and PowerBuilder components use a mix of derived EJB properties (since these component models are wrapped in EJBs), and properties that you can edit directly.

Before you migrate

Before migrating from EAServer 5.*x* to EAServer 6.0:

- Make a backup of your existing 5.*x* installation.
- Consider what you need to migrate. For example, exclude unused components, outdated samples, obsolete code, and so on, if no longer needed.

- If possible, migrate from a server you are not using for production. EAServer 5.x may need patches to ensure proper migration to EAServer 6.0. Patching the server may alter its behavior, and require the server to be restarted numerous times (after patch application, and after patch removal).
- Develop a migration strategy for clusters. If all servers in the cluster have the same components, you need only migrate the primary server, then use 6.0 clustering to synchronize with other 6.0 servers.
- Consider facilities that are no longer supported in EAServer 6.0, since they will either cause migration failures, or be marked as ineligible for migration. For example, PowerBuilder components with a version earlier than 10.0, CORBA/EJB Web Service Toolkit Web services, and OCI 8 or earlier connection caches are no longer supported in EAServer 6.0. See “Unsupported features” on page 4.
- Decide whether you will execute the migration process in GUI mode, or from the command line.
- Determine whether the 5.*x* and 6.0 server will be on the same machine, and whether the 5.*x* server will be running during the migration process. This determines whether the migration is performed in local mode or remote mode. Java Message Service (JMS) and Web service migration are not available in local mode.
- Check for packages with mixed component types, or applications that contain non-J2EE components. You must separate mixed-component packages, and non-J2EE packages from applications and migrate them separately. See “Packages with mixed component types” on page 15.
- Consider your migration strategy. You may want to migrate entities one at a time, checking successful migration of the single entity before moving on to the next entity. The same may apply for entity types. For example, you might want to migrate connection caches first, and ensure they have been migrated successfully. Then migrate components that use those connection caches.
- Determine whether there are dependencies in your 5.*x* server that may need to be manually migrated to 6.0 to allow successful migration. For example:
  - Are there Java classes in the system class path that may be needed for successful deployment?
  - Are there configuration or *ini* files that may be needed at runtime by the component you are migrating?

- Are there environment variables that may be needed by components?
- Do any of your components have dependencies on the JAGUAR environment variable? The run-server script sets a system property for the JAGUAR environment variable, and also sets it to the same value as DJC_HOME. This means any components running inside the server that depend on JAGUAR being set at the root of the server installation maintain the correct value. However, you should still consider how your components depend on the JAGUAR environment variable.For example, you must manually migrate the *$JAGUAR/foobar_app/config* directory to the *$DJC_HOME/foobar_app/config directory*, if *$JAGUAR/foobar_app/config* is not automatically migrated due to settings in your application that specify a dependency on this directory.

Additional requirements for EAServer 5.2 installations

The migration tool automatically applies required patches so you can safely migrate your EAServer entities to EAServer 6.0. To safely use the automated patch mechanism, your EAServer 5.2 installation must have the consolidated EBFs 12802 and 12803 applied. If your EAServer 5.2 installation does not have these patches applied, do one of the following:

1 Go to the Sybase Web site at http://sybase.com, download and install EBF numbers 12802 and 12803 to your server, then migrate using the automated patch mechanism.

2 Go to the Sybase Web site at http://sybase.com, download and install EBF numbers 13777 and 13778 to your server, then migrate, without using the automated patch mechanism.

3 Manually upgrade to EAServer 5.3, then migrate using the automated patch mechanism.

If the migration tool determines your EAserver installation is version 5.2, it displays a relevant message informing you to apply the patch or upgrade, similar to:

```
Do you want to patch the 5.x Server with a temporary
patch for migration? The patch corresponds to ebf# 13777
(Windows), 13778 (Solaris), 13779 (Linux). The patch can
be downloaded at: www.sybase.com.
```

For EAServer versions 5.0, 5.1, and 5.3, you can migrate directly to EAServer 6.0 using the automated patch mechanism.

Unsupported features

EAServer 6.0 is more stringent in terms of specification compliance. Entities you could successfully deploy to EAServer 5.*x* may not be deployable to 6.0 unless they meet specification requirements.

These items are no longer supported in EAServer 6.0 and cannot be migrated:

- Packages with mixed component types (J2EE, CORBA, and PowerBuilder). See “Packages with mixed component types” on page 15.
- Applications with non-J2EE component types.
- Java CORBA entities whose implementation classes are not in a Java package.
- PowerBuilder components with a version earlier than 10.0. See “CORBA and PowerBuilder packages” on page 17 for more information.
- OCI version 8 and lower.
- Web Service Toolkit CORBA or EJB Web services.

Manual migration

Some settings from 5.*x* are not migrated automatically. You must manually migrate:

- Memory usage limits
- Custom authentication service
- Custom role service
- Custom authorization service

Backward-compatibility issues

Migrating a component may fail if it contains something that is not backward-compatible, and has not been flagged for checking in the migration tool. For example, because of increased specification compliance a program entity deployed to EAServer 5.*x*, may not be allowed in EAServer 6.0. This is not detected until the entity is deployed to EAServer 6.0. You must update any such entities to meet the specification.

Example

The JSP compiler in EAServer 6.0 is more strict with syntax checking than EAServer 5.*x*. This example has an extra comma in the JSP import statement, which works in EAServer 5.x, but results in a compilation error in EAServer 6.0:

```
<%@ page import="java.io.IOException," %>
<%
out.println("This is a test");
%>
```

There may be other errors (with datatypes, for example) deployment failures because of backward-compatibility issues.

CHAPTER 2

Migrating to EAServer 6.0

While migrating EAServer 5.*x* to EAServer 6.0:

- The first step in the migration is to connect to the EAServer 5.*x* and 6.0 servers. Provide proper connection information, and verify you are connecting to the server you want to migrate. You are notified of any connection errors, and can also check the migration log file.
- When executed from the command line, migration is a three-step process:

  a Check

  b Export

  c Deploy

  However, in GUI mode, you can distinguish only between check and migrate. Migrate combines the export and deploy functions.

- The check strp scans the 5.*x* repository to see which entities can be migrated. There are various reasons an entity may not be eligible for migration. See “Unsupported features” on page 4 and “Backward-compatibility issues” on page 5. Each entity that cannot be migrated generates warning messages to both the screen and log file that include the reason the entity cannot be migrated.
- All entities within other entities (applications contain packages, packages contain components, and so on), must be eligible for migration.
- In some cases, you may be able to correct problems that prevent entities from being migrated. For example, you can upgrade your PowerBuilder version, connection cache version, and so on.
- When migrating entities one at a time, migrate entities with dependents first. For example, if you have an EJB that uses a connection cache for CMP, migrate the connection cache before you migrate the EJB.

  When migrating multiple entities, the migration process migrates dependent entities first. For example, if you choose to migrate both an EJB and the connection cache on which it depends at the same time, the connection cache is migrated first. You need to perform the migration in two separate steps.
- The export and deploy steps export the entity from 5.*x* (the exported entity may be an exported property file, exported J2EE archive file with the *sybase-easerver-config.xml* file, or exported package Jaguar JAR file.) Then the exported entity is deployed to 6.0
- Any errors generated during the export or deploy process display on both the screen (in GUI mode) and the migration log file.

Executing *migrate* from the command line

This section describes the migrate command line tool.

Description

Migrates J2EE applications, Web applications, Web services, EJBs, connectors, application clients, roles, configured queues, configured topics, queueConnectionFactories, topic connection factories, and other entities from EAServer 5.*x* to EAServer 6.0.

The migrate command has these restrictions:

- You cannot migrate individual application clients.

- You can migrate PowerBuilder packages, but not individual components. A package to be migrated cannot be of mixed component types.
- The ServletPersistentCache is not used in EAServer 6.0 and is not migrated.
- Oracle version 8 and earlier connection caches are not migrated.
- You can migrate an EAServer 5.*x* role if it contains at least one authorized user, or one excluded user. Digital IDs are not supported, and therefore, not migrated to EAServer 6.0. Also, the default roles from 5.*x* (Admin Role, Debug Role, ThreadManager) are migrated only if you have modified them.
- You can migrate CORBA, Java, and C++ components. However, you can migrate only packages, not individual components. You cannot migrate packages with mixed J2EE and CORBA components.
- You cannot migrate to a remote EAServer 6.0 installation.

Syntax

migrate <*-options*> <*-deploy_options*>

Option	Description
gui	If you use this option, which is the default, all other options are ignored, and migrate starts a graphical user interface with which to migrate your EAServer 5.*x* entities.
console	Runs the migration tool from the command line and allows other command line options. If you do not enter the -console option, migrate uses the -gui option.

entity	If you use this option, migrate allows you to migrate specific entities using the form EntityType:EntityName. For example:
	`migrate -console -entity ConnCache:Javacache`
	You can migrate more than one entity at a time. For example:
	`migrate -console -entity ConnCache:JavaCache -entity ConnCache:SampleCache`

eas5dir	The EAServer 5.*x* installation directory from which you are migrating. You must specify the full path to the EAServer 5.*x* directory or, if you do not, migrate uses the JAGUAR environment variable. If JAGUAR is empty, an error displays.
eas5outdir	A temporary directory used by the migrate command. The default location is *${djc.home}/migrate/<eas5host>-<eas5port>*. For example, if the *eas5host* is myhost and the *eaa5port* is 9000 and EAS 6.0 is installed at *d:\easerver6.0*, the default output directory is *d:\EAServer6.0\migrate\myhost-9000*.

Option	Description
eas5host	The host name on which the EAServer 5.*x* installation resides. The default is the value of the JAGUAR_HOST_NAME environment variable set in *bin/setenv.bat* (or *setenv.sh*) of the EAServer 5.*x* installation directory
eas5port	The EAServer IIOP port number used to connect to EAServer 5.*x*. The default is 9000.
eas5ServerName	The name of the EAServer 5.*x* server. For example, *Jaguar*.
eas5user	The name of the user connecting to EAServer 5.*x*. The default is jagadmin.
eas5password	The password of *eas5user*. The default is an empty password (no value).
eas5pin	The password/PIN used to access the PKCS 11 token. For example, *Sybase*.
host	The host name on which the EAServer 6.0 installation resides.
port	The EAServer IIOP port number used to connect to EAServer 6.0.
adminuser	The name of the administrative user connecting to EAServer 6.0.
servername	The name of the EAServer 6.0 server. For example, `new_server`.
httpport	The EAServer port number used to connect to EAServer 6.0. This is required for migrating EAServer 5.*x* Web services.
adminpwd	The password of *adminuser*.
keystorepass	The password of the EAServer 6.0 keystore. See the *EAServer Security and Administration Guide* for more information.
truststorepass	The password of the EAServer 6.0 truststore. See the *EAServer Security and Administration Guide* for more information.
jmsqueuemsgs	Migrates JMS message queue messages along with the queue.
doCheck true | false	Examines the EAServer 5.x installation to determine which entities are available to migrate. The default is true.
doExport true | false	Exports the entities to the migration output directory. The default is true.
doDeploy true | false	Deploys the exported entities to the EAServer 6.0 installation. The default is true.
overwrite	Overwrite the contents of the output directory location (true or false). The default is false.
verbose	Displays output messages of the migration command to the console.
quiet	Migration output messages are minimal.
help	Display help information.

Internally, the migration is performed in three steps:

1 Check – examines the EAServer 5.*x* server to determine the entities available to migrate.

2 Export – the EAServer 5.*x* entities are exported to the migration output directory.

3 Deploy – exported entities are deployed individually to the destination EAServer 6.0 installation.

Note When *overwrite* is true at the "Check" step, all files in the migration output directory are deleted.

When *overwrite* is true at the "Deploy" step, deploying an entity fails if the entity already exists in the target EAServer 6.0 installation.

After running migrate, check the *migrate.log* file for error and informational messages. *migrate.log* is located in the EAServer 6.0 *logs* subdirectory.

Examples

This example migrates an existing EAServer 5.*x* repository, identified by the JAGUAR environment variable, to the current EAServer 6.0 installation with overwrite set to true:

```
migrate -console -overwrite true
```

This example migrates from a specific EAServer 5.*x* directory, using the doExport option, then uses doDeploy to deploy to an EAServer 6.0 installation:

```
migrate -console -eas5dir f:\sybase\jag52025p5 -
eas5host testhost -eas5port 9000 -eas5user jagadmin -
doExport -eas5outdir c:temp\migrate1
```

```
migrate -console -doDeploy -eas5outdir c:temp\migrate1
```

```
2005-09-16 14:08:52.210 INFO  main [MigrateTool]
Migrated archive files are in c:emp\migrate1
2005-09-16 14:08:52.835 INFO  main [DeployTool]
Extracting files from archive: C:
emp\migrate1\HelloWorldApp.ear
```

Using migrate with the -gui option

The migrate command line tool includes a -gui option, which allows you to use a graphical user interface to migrate EAServer 5.*x* entities to EAServer 6.0. If you use the -gui option , which is the default (or no options), all other options are ignored.

❖ **Using migrate -gui**

1 From a command line, enter migrate from the EAServer 6.0 *bin* directory (*%djc_home%\bin* on Windows or *$djc_home/bin* on Solaris).

2 Select File | Connect. The Connection Parameters window opens. Enter these parameters, which indicate the locations of your EAServer 5.*x* and EAServer 6.0 installations and click Connect:

- EAServer 5.*x* information:
  - Location – the complete path to the EAServer 5.x installation. The default is defined by the JAGUAR variable.
  - Username – the user name used to connect to EAServer 5.*x*. The default is "jagadmin."
  - Password – the password of the user who is connecting to EAServer 5.*x*. The default is blank (that is, no password).
  - Hostname – the host name on which EAServer 5.*x* resides. The default is "localhost."
  - IIOP Port – EAServer 5.*x* IIOP port number to which you are connecting. The default is 9000.
  - HTTP Port – EAServer 5.*x* HTTP port number to which you are connecting. The default is 8080.
  - Server – the 5.*x* server to which you are connecting. The default is "Jaguar."
  - PKCS11 Pin – the password/PIN used to access the EAServer token used by Security Manager. For example, Sybase.
- EAServer 6.0 information:
  - Location – the complete path to the EAServer 6.0 installation. The default is *%DJC_HOME%\bin* on Windows or *$DJC_HOME/bin* on Solaris.
  - Username – the user name used to connect to EAServer 6.0. For example, admin@system.

- Password – the password of the admin@system user who is connecting to EAServer 6.0. This password was established during installation of EAServer 6.0. It can be set using the set-admin-password batch file located in the *bin* subdirectory of your EAServer 6.0 installation.
- Hostname – the host name on which EAServer 6.0 resides. The default is host name of the server.
- IIOP Port – EAServer 6.0 IIOP port number to which you are connecting. The default is 2000.
- HTTP Port – EAServer 6.0 HTTP port number to which you are connecting. The default is 8000.
- Keystore password – the password of the EAServer 6.0 keystore. See the *EAServer Security and Administration Guide* for more information.
- Truststore password – the password of the EAServer 6.0 truststore. See the *EAServer Security and Administration Guide* for more information.

3 Select File | Scan EAServer 5.x Repository. The migration tool scans the EAServer 5.x installation for entities that can be migrated to EAServer 6.0, which may take a couple of minutes, depending on the size of the repository. The migration tool displays a message when the scanning completes.

4 Highlight (or expand) the various entities (Application, ConnCache, Connector, JAAS configuration, clusters, and so on) and select the entities that you want to migrate. You can select either:

- Select All Entities – migrates all entities. This option is the default.
- Unselect All Entities – unselects all items.

  The effect of “Select All Entities”or “Unselect All Entities” differs depending on which node you have selected in the tree view. If you select the root node of “Repository” then selecting Select All or Unselect All selects or unselects all the entities. However, if you selected another node, then only the entities under that node are selected or unselected. For example, if you select All Entities when the ConnCache Node is selected, all of the Conncaches are checked, and therefore migrated. The check boxes for all of the other entities are unaffected.

5 Select File | Migrate to migrate the selected EAServer 5.*x* entities to your EAServer 6.0 installation. The migration tool displays a message when the migration completes.

6 Overwrite existing files in the migration directory – use this option when you run migrate from the same *eas5host* and *eas5port* more than once. In this scenario some files are written to the output directory during the initial migration. If you run migrate -gui again, the connection fails and displays the message "Overwrite flag is false but there are files in the output directory: <the output directory>." Close the Connect dialog, return to the main window, and select Overwrite. Then connect again.

7 Prompt to apply and remove EAServer 5.x migration patches – You must apply patches to your EAServer 5.x installation as part of the migration process by selecting this checkbox. See "Patching EAServer 5.x" on page 14 for more information.

Note You must run the migration tool (both -gui and -console options) on the machine on which your EAServer 6.0 installation resides. You cannot run it remotely. Also, an EAServer 5.*x* installation must reside on the same machine, since the migration process uses some commands included with that installation. However, you can migrate entities from an EAServer 5.*x* installation that resides on a different machine.

Patching EAServer 5.x

There are several patches you must apply to EAServer version 5.x for migration to successfully complete. See "Additional requirements for EAServer 5.2 installations" on page 3 for more information about the individual patch numbers, and how to download them. If you want EAServer to automatically apply required patches during migration, select "Automatically Apply Patches to EAServer" when you are entering parameters in the migrate GUI.

When the migration has finished, select “Automatically Remove Applied Patches from EAServer 5.x.” If you do not, you may encounter problems when performing other operations on your EAServer 5.*x* server.

Note The migration patch does not affect export/deployment from Jaguar Manager in EAServer 5.*x*, since it has its own class path, which is not affected by the server’s classpath being switched by the migration patching process.

Packages with mixed component types

You cannot directly migrate packages with mixed component types. If a package contains both EJB and non-EJB components (for example, C++ CORBA, Java CORBA, and PowerBuilder) you must create separate packages that contain either only EJB components or only non-EJB components, then migrate those packages.

For example, let’s say you have a package called “Mixed” that contains an EJB component “ejbcomp” and a Java CORBA component called “jccomp”. To migrate both of these components:

1 In EAServer 5.*x*, start EAServer Manager.

2 Navigate to the Packages folder.

3 Right-click Packages and select New Package.

4 Name the new package “MovedComps”.

5 Click the package named “Mixed” in the tree view.

6 Right-click jcccomp.

7 Choose Copy from the menu.

8 Right-click the package “MovedComps” in the tree view.

9 Choose Paste, which copies “jcccomp” into the “MovedComps” package.

10 Select “jcccomp” in the tree view under the “Mixed” package.

11 Right-click and select Remove, which removes “jcccomp” from the “Mixed” package.

You have now split the components into two packages, and you can migrate each package.

CORBA components

Migrating applications that contain CORBA packages

To migrate applications that contain CORBA packages:

1. Use the migration tool with the -gui option to migrate individual CORBA packages, *not* as part of an application. The CORBA packages are listed separately from the application to which they belong. See "Using migrate with the -gui option" on page 12.
2. Temporarily modify the application to *not* include the CORBA packages so that the application only contains J2EE entities. Then migrate the application as a J2EE application.

   Modify the application using one of these methods:

   - EAServer Manager – browse to Application | AppName | Packages | PackageName, right-click the appropriate package and select Remove.
   - jagtool – run:

     ```
     jagtool remove Package:PackageName Application:AppName
     ```

     where *PackageName* is the name of the package and *AppName* is the name of the application you are removing.
   - Edit the *.properties* file – modify the property com.sybase.djc.application.packages, and remove the CORBA package from the list of packages.
3. After migrating the application to EAServer 6.0, modify it so that it again contains the CORBA packages. Use one of:

   - Management Console – browse to Application | AppName | Packages. Right-click Packages and select Install Package, then choose the CORBA package that had been removed.
   - jagtool – run:

     ```
     jagtool install Package:PackageName Application:AppName
     ```

     where *PackageName* is the name of the package and *AppName* the name of the application you are installing.

- Edit the *.properties* file – modify the property com.sybase.djc.application.packages, and add the CORBA package to the list of packages.

Note You cannot directly migrate Java CORBA components that have an implementation class that is not in a package, or manually create a Java CORBA component in EAServer 6.0 that has an implementation class that is not in a package.

If you have this situation, move the implementation class into a Java package, and change the component properties to reflect this. That is, modify com.sybase.jaguar.component.java.class.

CORBA and PowerBuilder packages

The migration tool exports the CORBA and PowerBuilder packages as Jaguar JAR files, then deploys them to EAServer 6.0. Here is how the entries in a Jaguar JAR file are processed during deployment:

- Any class files in the JAR under *$JAGUAR/java/classes* or *$JAGUAR/html/classes* are copied to *$DJC_HOME/java/classes*, *$DJC_HOME/html/classes* respectively.
- Any JAR files inside the JAR are copied to *$JAGUAR/java/classes*.
- You cannot migrate PowerBuilder components with a version earlier than 10.0. If you have such a component:
  a Migrate the PowerBuilder component from the current version of the PowerBuilder IDE to version 10.0 or later.
  b Redeploy the PowerBuilder component to EAServer version 5.*x*.
  c Migrate the component (contained in a package) to EAServer 6.0.

CORBA C++ components

To build a C++ component in EAServer 6.0:

1 Create the IDL and place it in the EAServer 6.0 IDL repository directory (*%djc_home%\Repository\IDL* on Windows or *$djc_home/Repository/IDL* on Solaris).

2 Create the component properties. In most cases, the property settings from EAServer 5.*x* work. Use the */pb-server/config/pb-debug-broker.xml* file located within the *djc_home* directory as a sample.

3 Invoke the EAServer compiler script (*%djc_home%\bin\djc.bat* on Windows or *$djc_home/bin/djc.sh* on Solaris) on the package/component name. This generates IDL stubs and skeletons, and the necessary EJB proxies that invoke the CORBA C++ component.

Web services and Web applications

You can migrate 5.*x* Web services to EAServer 6.0 using either the GUI or the Management Console.

Restrictions

Keep these restrictions in mind when migrating Web services:

- If a Web application contains a Web service, it displays as part of a Web service collection, not as part of the Web application.
- Web services are migrated at the collection level, not the individual Web service level.
- Only POJO (plain old Java object) type Web services are migrated. EJB and CORBA Web services are not migrated.

  If you have EJB or CORBA Web services, you can:

  a Migrate the EJB or CORBA component to 6.0.

  b Use Web Services Toolkit in EAServer 6.0 to expose the EJB or CORBA component (wrapped as an EJB) as a new Web service.

Prerequisites

Before you can migrate 5.*x* Web services, make sure that:

- The Web administration console is installed.

- EAServer 5.*x* from which you are migrating is running and the Web Services Toolkit is installed.
- EAServer 6.0 is running.

If you run migrate from the console, you must specify these additional options:

- -eas5httpport – EAServer 5.*x* HTTP port number.
- -httpport – EAServer 6.0 HTTP port number.

In GUI mode, select the WebService service type. After scanning the EAServer 5.*x* repository, all 5.*x* POJO WebService collections display.

Role mapping

When migrating a Web application that has role mappings, if the J2EE role is different from the mapped EAServer role, a third role with the name of the J2EE role name is created. The generated configuration is similar to this:

```
<setProperties securityDomain="default">
   <addRole role="eaf.EAF Admin" toRole="Admin Role"/>
    <addRole role="eaf.EAF Admin" toRole="EAF Admin"/>
  </setProperties>
```

Service components

Service components are migrated automatically.

If you only want to migrate a component (and do not want to install it as a service component in EAServer 6.0), remove the component from the service components in EAServer 5.*x* before migrating.

To manually migrate service components from EAServer 5.*x* to EAServer 6.0:

1 To mark something as a service in 6.0, create an instance of a com.sybase.djc.server.ServiceComponent and add it to the application service property serviceComponents.

2 Migrate the component. For example, “TestService/SimpleService” from 5.*x* to 6.0.

The migrated component is an EJB wrapper with a package named `ejb.components.testservice`.

3 Start the server. Check the log file to verify the service has started, as this sample illustrates:

```
Server Log File: M:\target1.4\logs\HOST-W2K.log
HTTP Listeners: http://HOST-W2K:8000,https://HOST-W2K:8001
IIOP Listeners: iiop://HOST-W2K:2000
Starting services...
DEBUG: In SimpleService.start
Accepting Connections: iiop://HOST-W2K:2000
Accepting Connections: http://HOST-W2K:8000
Accepting Connections: https://HOST-W2K:8001
Server Started
DEBUG: In SimpleService.run
The Management Console can be accessed at http://HOST-W2K:8000/console
```

Connection caches

EAServer 5.*x* connection caches have been removed from EAServer 6.0 and replaced with *Datasources*, which wrap native and JDBC connections.

Here is a list of the EAServer 5.*x* sample connection caches and the equivalent EAServer 6.0 data source, if any:

Connection cache	Datasource equivalent
EASDemo	None.
EASDemo_jConnect	None.
EASDemo_JdbcOdbc	None.
JavaCache	Default.
Add a predefined JavaCache detasource as an alias for “default”.	The migrated “default” datasource refers to the predefined connection cache (which is the primary database), and sets the default properties of the JavaCache according to your main database requirements.
SampleCache	None.
ServletPersistenceCache	DistributedWebApp.

Note The EAServer 6.0 repository does not include other data source definitions. Connection caches are migrated from EAServer 5.*x* to EAServer 6.0. There is also a *default-data-sources.xml* configuration file located in the *$djc_home/config* on Solaris or *%djc_home%\config* on Windows directory that contains the default datasource definitions as well as various commented out samples.

Native connection caches

EAServer 5.*x* native connection caches are replaced in EAServer 6.0 with JDBC driver/connection wrappers. To configure them, create an instance of the JDBC wrapper. Samples are located in *$djc_home/config/default-data-sources.xml*.

The serverName property represents the name that the underlying native API uses to resolve the server name and location. For example:

- JCM_Odbc – serverName represents the DSN.
- JCM_Sybase – serverName represents the DSQUERY entry in the sql.ini or the interfaces file for Open Client™.
- JCM_Oracle – serverName represents the *tnsnames.ora* service name.

Here is a list of the connection library name changes from EAServer 5.*x* to EAServer 6.0:

EAServer 5.x library	EAServer 6.0 library
ODBC	JCM_Odbc
ODBCU	JCM_Odbc_Unicode
OCI_7	Not supported in EAServer 6.0
OCI_8	Not supported in EAServer 6.0
OCI_9	JCM_Oracle
OCI_10	JCM_Oracle
OCI_9U	JCM_Oracle_Unicode
OCI_10U	JCM_Oracle_Unicode
CTLIB_110	JCM_Sybase

Java archive files

You can migrate your Java archive files from 5.*x* to 6.0, including third-party JAR files (.jar and .zip) from *$JAGUAR/java/lib*, and extensions JAR files from *$JAGUAR/extensions*:

1 Within the JavaArchive foler are the Server and Extensions folders, which display all available archives for migration. Select the archive files you want to migrate.

2 Select File | Migrate to migrate the selected EAServer 5.*x* entities to your EAServer 6.0 installation.

Query caching

EAServer 5.*x* query caching settings are not automatically migrated to EAServer 6.0. If query caching in EAServer 5.*x* is detected, a warning message is printed in the deployment and migration logs, similar to:

```
2006-06-02 07:55:51.803 WARN  main [EjbCompiler] query
findByPrimaryKey in EJB BusinessStatus used Query
Caching in 5.x,  manual migration is required to setup
Query Caching for this Query in 6.0.
```

You must manually set these three attributes for the queryMethod in the *ejbjar-config* file to enable query caching:

- cacheSize – the maximum number of query results that will be held in cache. Each query result may consist of zero or more database rows. To avoid overflow, rows are removed from cache using a least recently used (LRU) discard strategy.
- cacheTimeout – the maximum number of seconds that a result stored in cache is considered valid. Do not use this property to attempt to control the cache size, because invalid rows are discarded from cache when an attempt is made to access them.
- isolationLevel – a logical isolation level for this query. A logical isolation level is distinct from the SQL isolation level that is applied to any JDBC database connections that are participating in the transaction. Typically, the SQL isolation level is ReadCommitted, and for some entity classes or query methods a higher isolation level is requested. Use one of the *_WITH_CACHE isolation levels.

For more information, see the QueryMethodProperty class documentation, which is available from the Web Management Console at *http://hostname:8000/javadocs/public/index.html*. Where *hostname* is the name of the host on which your EAServer 6.0 installation resides.

Note EAServer 5.x supports named caches, while EAServer 6.0 does not. In EAServer 6.0, you must specify the cache properties for each query.

Configuring table-mapping properties

EAServer 5.*x* allows you to configure table-mapping properties for entity components. These table-mapping settings do not automatically migrate to EAServer 6.0. Manually migrate the table- mapping settings by setting up these three PersistentObject properties in the *ejbjar-config* file:

- SetdeleteStatement
- SetinsertStatement
- SetupdateStatement

For more information, see the PersistentObjectProperty class documentation, which is available from the Web Management Console at *http://hostname:8000/javadocs/public/index.html*, where *hostname* is the name of the host on which your EAServer 6.0 installation resides.

If you used table-mappings in your EAServer 5.*x* installation, you must also manually update the queries in your EAServer 6.0 *ejb-jar* configuration file to refer to the multiple tables being used.

Note In EAServer 5.*x* custom SQL queries were allowed to return only the key fields for the entity, but in EAServer 6.0 all fields (both key fields, and state fields) must be returned by the query. This also requires manual updating after migration.

EJB-QL support

This section describes the differences in EJB QL query support between EAServer 5.x and EAServer 6.0. See the *Enterprise JavaBeans Specification, version 2.0* (or *2.1*) for complete information about EJB-QL.

These situations do not directly effect EAServer migration, but you must keep these differences in mind when using the <ejb-ql> query tag in EAServer 6.0:

- In EAServer 5.*x*, you could deploy an EJB JAR that had an ejb-ql query that used the abstract-schema-name to refer to various EJBs in the query. The case (uppercase or lowercase) of the abstract-schema-name in the ejb-ql query was not required to match the case of the EJB's abstract-schema-name.

  EAServer 6.0 is specification-compliant and requires the case in both names to match, otherwise deployment fails, and a NoClassDefFound Exception is generated. This example illustrates a case that is acceptable in EAServer 5.*x*, but fails in EAServer 6.0:

```
<ejb-name>ProcessType</ejb-name>
   <abstract-schema-name>ProcessType</abstract-schema-name>
...
      <query>
        <query-method>
          <method-name>findAll</method-name>
          <method-params />
        </query-method>
        <ejb-ql>SELECT OBJECT(a) FROM Processtype AS a</ejb-ql>
      </query>
```

  The query uses "Processtype" with a lowercase "t", while the abstract-schema-name uses "ProcessType" with an uppercase "T".

- In EAServer 5.*x*, you could specify an empty <ejb-ql> element for a query in an EJB.

  If you specify an empty <ejb-ql> element in EAServer 6.0, the deployment finishes successfully, but the SQL entry for the query in the EJB JAR configuration file is changed to "???", signifying there is no SQL for the query. You must provide SQL for the query, and run recompile on the EJB. If not, an error similar to this is entered in the log file:

```
[loadComponents] 2006-06-02 07:56:08.126 WARN  main
[PersistenceManager:ejb.components.gwfserver.TaskType] SQL statement is
missing for query method findAllChildTasks
```

  A normal ejb-ql query in the *ejbjar-config* file looks like this:

```
<queryMethod method="findAll()"
        ejbql="SELECT OBJECT(a) FROM BusinessStatus AS a"/>
```

  but if the <ejb-ql> element was empty in the deployment descriptor, you see:

```
<queryMethod method="findAllChildTasks(java.lang.Long p1,
java.lang.String p2)"
        sql="???"/>
```

  Here is an example of an empty <ejb-ql> element:

```
<query>
      <query-method>
          <method-name>findMandatoryChildTasks</method-name>
          <method-params>
             <method-param>java.lang.Long</method-param>
             <method-param>java.lang.String</method-param>
           </method-params>
      </query-method>
      <ejb-ql />
  </query>
```

- In EAServer 5.*x*, you could specify the ejb-ql for a query that contained double quotes, but if you specify the ejb-ql with double quotes in EAServer 6.0, an error is generated because double quotes are not specification-compliant.

  If you deploy an ejb-ql that contains double quotes in EAServer 6.0, the double quotes are automatically replaced with single quotes, and a warning message similar to this displays:

```
2006-06-02 07:52:14.040 WARN  main [EjbCompiler] ejbql for
findInactiveTasks in EJB TaskInstance contains double quotes which is not
spec compliant, replacing double quotes with single quotes.
```

You can still use the double quotes, since the ejb-ql is turned into a SQL query that is appropriate for the database that is being used for the container managed persistence.

JAAS configurations

When you migrate your JAAS configuration from EAServer 5.*x* to EAServer 6.0. The migration tool:

1 Checks EAServer 5.*x* for a JAAS configuration file.

2 Creates an entity that contains a JAAS configuration file, exports, then extracts the contents to create a local copy of the 5.*x* JAAS configuration file.

3 Merges the contents of the EAServer 5.*x* JAAS configuration file with the EAServer 6.0 JAAS configuration file.

4 Sets the properties for a default security domain to a login method of JAAS, and sets the appropriate JAAS section from the EAServer 5.*x* settings. See Chapter 10, "Security Configuration Tasks," in the *EAServer Security Administration and Programming Guide* for more information.

Certificates

If you migrate security certificates contained in the PKCS11 module of your EAServer 5.*x* installation, you must:

1 Provide the PKCS11 PIN for the EAServer 5.*x* server. If you do not provide a PIN, the default value "sybase" is used. To specify the PKCS11 PIN on the command line in console mode, use the -eas5pin option. The GUI includes a password field in the connect dialog that allows you to specify the PIN.

2 Provide the keystore password, and truststore password for the EAServer 6.0 server to which you are migrating. If you do not provide the passwords the default value "changeit" is used. The console options are -keystorepass and -truststorepass. The GUI includes fields for you to specify the passwords in the connect dialog.

3 Specify either -entity all (all entities, including security certificates), are migrated or -entity Security (if you specify single entities) when using migrate from the command-line.

Additional files

This section describes how additional files and extra files properties are handled during migration. When an entity has extra file properties specified, those extra files are packaged in the exported Jaguar JAR file. However, it is not included with the standard J2EE archive files (EARs, WARs, EJB JARs). Instead the migration process (in addition to migrating the J2EE archives), migrates the Jaguar JAR for J2EE entities and extracts the extra files. Keep these details in mind when migrating extra files:

- Files specified with an absolute name – copies the files only when migrating from a remote EAServer 5.*x* server.
- Files beginning with "../" – indicates relative to an EAServer 5.*x* installation directory. These files are copied to the EAServer 6.0 installation directory.
- Files beginning with "cpplib/" – copied to the EAServer 6.0 *cpplib* subdirectory.
- Files beginning with "java/classes" – copied to the EAServer 6.0 *genfiles/java/classes* subdirectory.
- Files beginning with an environment variable denoted by ${...} – copied to the corresponding location according to the environment variable if the contents are different then the existing files.

JMS messages

For JMS messages, only queue messages are migrated:

1 From the migration GUI, navigate to the folder "JMS/OPTIONS".

2 Select Migrate Queue Messages. The option is disabled by default.

If you are using migrate from the command line, use the -jmsqueuemsgs option to migrate the queue messages.

Migrating the demonstration application

Before you migrate the Car demonstration application from EAServer 5.*x*, you must patch EAServer 5.*x* with patches provided on your EAServer 6.0 installation CD:

- When you connect to EAServer using the migration GUI, a dialog asks you whether to install the EAServer 5.*x* patches.
- When you disconnect from EAServer, a dialog asks you whether to uninstall the EAServer 5.*x* patches.

CHAPTER 3

Post-migration tasks

After you migrate

After you migrate to EAServer 6.0:

- Restart the server to activate new bindings and settings for migrated entities.
- Ensure that all migrated entities run successfully.
- Consider shared resources such as databases. You have migrated to EAServer 6.0, but if you leave the version 5.*x* server running, both versions of the application may be running against the same resource.
- To use the same port numbers as 5.*x*, you must shut down EAServer 5.*x* for your listener profiles, change your EAServer 6.0 port numbers, and restart the 6.0 server.
- Synchronize any cluster migrated from 5.*x*.
- Manually migrate entities as necessary. See “Manual migration” on page 4.
- Configure the migrated entities as nesessary.
- Modify class loader settings to include non-J2EE packages that were removed from applications before migrating.
- Remove the *jdbc2_0-stdext.jar* file from the *lib* subdirectory of your EAServer 6.0 installation. EAServer 6.0 is J2EE 1.4 compliant and supports JDBC 3.0. Removing this file helps avoid ClassCastException errors.

Troubleshooting

You cannot migrate the Jaguar1 package from EAServer 5.x to EAServer 6.0:

Parser error messages

Because EAServer 6.0 uses Xerces as its XML parser (which parses the deployment descriptor), error messages from Xerces might not be particularly helpful in isolating problems. The following scenarios describe what the underlying problem causing some of these error messages might be.

Bad schema location

This is a deployment descriptor with an invalid schema reference:

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd1"
```

If you deploy a Web application with such a descriptor, you are likely to generate error messages similar to:

```
2006-03-31 13:46:22.079 INFO main [WebCompiler] Extracting files from archive:
D:\xfsong\workhome\nbprojects\WebTest\dist\WebTest.war
SAXParserException: org.xml.sax.SAXParseException: cvc-elt.1: Cannot find the
declaration of element 'web-app'.
Column Number: + 17
Line Number: + 6
2006-03-31 13:46:37.191 ERROR main [XmlDocument] An error occurred during
parsing.
2006-03-31 13:46:37.207 WARN main [DeployTool] Deploy module failed due to
exception: com.sybase.djc.xml.XmlException: An error occurred during parsing.
2006-03-31 13:46:37.207 ERROR main [DeployTool] Failed to deploy module:
D:\xfsong\workhome\nbprojects\WebTest\dist\WebTest.war
2006-03-31 13:46:37.207 INFO main [DeployTool] Running configuration script:
webapp-webtest
D:\sybase\60069\bin\undeploy.bat webapp-webtest
2006-03-31 13:46:37.711 WARN main [UndeployTool] Warning: No config file found
for webapp-webtest
Warning: No config file found for webapp-webtest
2006-03-31 13:46:37.821 INFO main [UndeployTool] Undeployed: webapp webtest
Undeployed: webapp webtest
```

This indicates that the XML parser does not recognize any of the XML elements, such as “web-app”.

Bad DTD location

This is a Web application with a deployment descriptor that has an invalid DTD definition:

```
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_31.dtd">
```

When you deploy a Web application with an invalid DTD definition, you see messages similar to:

```
2006-03-31 13:51:30.968 INFO main [WebCompiler] Extracting files from archive:
D:\xfsong\workhome\nbprojects\WebTest13\dist\WebTest13.war
2006-03-31 13:51:38.189 WARN main [DeployTool] Deploy module failed due to
exception: com.sybase.djc.xml.XmlException
2006-03-31 13:51:38.189 ERROR main [DeployTool] Failed to deploy module:
D:\xfsong\workhome\nbprojects\WebTest13\dist\WebTest13.war
2006-03-31 13:51:38.204 INFO main [DeployTool] Running configuration script:
webapp-webtest13
D:\sybase\60069\bin\undeploy.bat webapp-webtest13
2006-03-31 13:51:38.785 WARN main [UndeployTool] Warning: No config file found
for webapp-webtest13
Warning: No config file found for webapp-webtest13
2006-03-31 13:51:38.927 INFO main [UndeployTool] Undeployed: webapp webtest13
Undeployed: webapp webtest13
```

This message does not clearly identify the problem. Re-run deploy using the "-verbose" option, which produces the following error messages that clearly shows the root cause of the error to be the missing DTD file:

```
at com.sybase.djc.xml.XmlDocument.read(XmlDocument.java:197)
at
com.sybase.djc.deploy.webapp.WebCompiler.getXmlDocument(WebCompiler.java:1244
)
at com.sybase.djc.deploy.webapp.WebCompiler.run(WebCompiler.java:864)
at com.sybase.djc.deploy.DeployTool.run(DeployTool.java:783)
at com.sybase.djc.deploy.DeployTool.start(DeployTool.java:261)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.
java:25)
at java.lang.reflect.Method.invoke(Method.java:324)
at
com.sybase.djc.bootstrap.StartApplication.main(StartApplication.java:250)Caus
ed by: java.io.FileNotFoundException: http://java.sun.com/dtd/web-app_2_31.dtd
at
```

```
sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.
java:812)
at org.apache.xerces.impl.XMLEntityManager.setupCurrentEntity(Unknown Source)
at org.apache.xerces.impl.XMLEntityManager.startEntity(Unknown Source)
at org.apache.xerces.impl.XMLEntityManager.startDTDEntity(Unknown Source)
at org.apache.xerces.impl.XMLDTDScannerImpl.setInputSource(Unknown Source)
at
org.apache.xerces.impl.XMLDocumentScannerImpl$DTDDispatcher.dispatch(Unknown
Source)
at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanDocument(Unknown
Source)
at org.apache.xerces.parsers.XML11Configuration.parse(Unknown Source)
at org.apache.xerces.parsers.XML11Configuration.parse(Unknown Source)
at org.apache.xerces.parsers.XMLParser.parse(Unknown Source)
at org.apache.xerces.parsers.AbstractSAXParser.parse(Unknown Source)
at org.apache.xerces.jaxp.SAXParserImpl$JAXPSAXParser.parse(Unknown Source)
at com.sybase.djc.xml.XmlDocument.read(XmlDocument.java:184)
```

Deployment descriptor validation error

If you have a Web application named “WebTest”, but “name-display” is not allowed by the schema, you see validation error messages similar to:

```
2006-03-31 14:02:55.365 INFO main [WebCompiler] Extracting files from archive:
D:\xfsong\workhome\nbprojects\WebTest\dist\WebTest.war
SAXParserException: org.xml.sax.SAXParseException: cvc-complex-type.2.4.a:
Invalid content was found starting with element 'name-display'. One of
'{"http://java.sun.com/xml/ns/j2ee":description,
"http://java.sun.com/xml/ns/j2ee":display-name,
"http://java.sun.com/xml/ns/j2ee":icon,
"http://java.sun.com/xml/ns/j2ee":distributable,
"http://java.sun.com/xml/ns/j2ee":context-param,
"http://java.sun.com/xml/ns/j2ee":filter,
"http://java.sun.com/xml/ns/j2ee":filter-mapping,
"http://java.sun.com/xml/ns/j2ee":listener,
"http://java.sun.com/xml/ns/j2ee":servlet,
"http://java.sun.com/xml/ns/j2ee":servlet-mapping,
"http://java.sun.com/xml/ns/j2ee":session-config,
"http://java.sun.com/xml/ns/j2ee":mime-mapping,
"http://java.sun.com/xml/ns/j2ee":welcome-file-list,
"http://java.sun.com/xml/ns/j2ee":error-page,
"http://java.sun.com/xml/ns/j2ee":jsp-config,
"http://java.sun.com/xml/ns/j2ee":security-constraint,
"http://java.sun.com/xml/ns/j2ee":login-config,
"http://java.sun.com/xml/ns/j2ee":security-role,
"http://java.sun.com/xml/ns/j2ee":env-entry,
```

```
"http://java.sun.com/xml/ns/j2ee":ejb-ref,
"http://java.sun.com/xml/ns/j2ee":ejb-local-ref,
"http://java.sun.com/xml/ns/j2ee":service-ref,
"http://java.sun.com/xml/ns/j2ee":resource-ref,
"http://java.sun.com/xml/ns/j2ee":resource-env-ref,
"http://java.sun.com/xml/ns/j2ee":message-destination-ref,
"http://java.sun.com/xml/ns/j2ee":message-destination,
"http://java.sun.com/xml/ns/j2ee":locale-encoding-mapping-list}' is expected.
Column Number: + 19
Line Number: + 7
2006-03-31 14:02:57.715 ERROR main [XmlDocument] An error occurred during
parsing.
2006-03-31 14:02:57.715 WARN main [DeployTool] Deploy module failed due to
exception: com.sybase.djc.xml.XmlException: An error occurred during parsing.
2006-03-31 14:02:57.715 ERROR main [DeployTool] Failed to deploy module:
D:\xfsong\workhome\nbprojects\WebTest\dist\WebTest.war
2006-03-31 14:02:57.715 INFO main [DeployTool] Running configuration script:
webapp-webtest
D:\sybase\60069\bin\undeploy.bat webapp-webtest
2006-03-31 14:02:58.326 WARN main [UndeployTool] Warning: No config file found
for webapp-webtest
Warning: No config file found for webapp-webtest
2006-03-31 14:02:58.467 INFO main [UndeployTool] Undeployed: webapp webtest
Undeployed: webapp webtest
```

Bad closing tag in deployment descriptor

If you have a Web application named “WebTest”, that contains a bad closing tag in the deployment descriptor, you see error messages similar to:

```
2006-03-31 14:08:27.315 INFO main [WebCompiler]
Extracting files from archive:
D:\xfsong\workhome\nbprojects\WebTest\dist\WebTest.war
SAXParserException: org.xml.sax.SAXParseException: The
end-tag for element type "display-name" must end with a
'>' delimiter.
Column Number: + 40
Line Number: + 7
2006-03-31 14:08:29.493 WARN main [DeployTool] Deploy
module failed due to exception:
com.sybase.djc.xml.XmlException
2006-03-31 14:08:29.493 ERROR main [DeployTool] Failed
to deploy module:
D:\xfsong\workhome\nbprojects\WebTest\dist\WebTest.war
2006-03-31 14:08:29.493 INFO main [DeployTool] Running
configuration script: webapp-webtest
```

```
D:\sybase\60069\bin\undeploy.bat webapp-webtest
2006-03-31 14:08:30.166 WARN main [UndeployTool]
Warning: No config file found for webapp-webtest
Warning: No config file found for webapp-webtest
2006-03-31 14:08:30.291 INFO main [UndeployTool]
Undeployed: webapp webtest
Undeployed: webapp webtest
```

Missing closing tag in the deployment descriptor

If you have a Web application named “WebTest”, that is missing a closing tag in the deployment descriptor, you see error messages similar to:

```
2006-03-31 14:14:09.750 INFO main [WebCompiler]
Extracting files from archive:
D:\xfsong\workhome\nbprojects\WebTest\dist\WebTest.war
SAXParserException: org.xml.sax.SAXParseException: The
element type "display-name" must be terminated by the
matching end-tag "".
Column Number: + 3
Line Number: + 18
2006-03-31 14:14:11.943 WARN main [DeployTool] Deploy
module failed due to exception:
com.sybase.djc.xml.XmlException
2006-03-31 14:14:11.943 ERROR main [DeployTool] Failed
to deploy module:
D:\xfsong\workhome\nbprojects\WebTest\dist\WebTest.war
2006-03-31 14:14:11.943 INFO main [DeployTool] Running
configuration script: webapp-webtest
D:\sybase\60069\bin\undeploy.bat webapp-webtest
2006-03-31 14:14:12.664 WARN main [UndeployTool]
Warning: No config file found for webapp-webtest
Warning: No config file found for webapp-webtest
2006-03-31 14:14:12.789 INFO main [UndeployTool]
Undeployed: webapp webtest
Undeployed: webapp webtest
```

Invalid comments

If a JSP file contains a comment:

```
 <!-- sy:debug>
</sy:debug -->
```

EAServer 5.x allows users to deploy J2EE archives that contain incorrect deployment descriptors or comments. However, such a file will neither compile in nor successfully migrate to EAServer 6.0.

Content format exception in the deployment descriptor

If you have a Web application named “WebTest” that has a format exception in the deployment descriptor, you see error messages similar to:

```
2006-03-31 14:24:53.643 INFO main [WebCompiler]
Extracting files from archive:
D:\xfsong\workhome\nbprojects\WebTest\dist\WebTest.war
SAXParserException: org.xml.sax.SAXParseException:
cvc-complex-type.2.4.a: Invalid content was found
starting with element 'a'. One of
'{"http://java.sun.com/xml/ns/j2ee":description,
"http://java.sun.com/xml/ns/j2ee":display-name,
"http://java.sun.com/xml/ns/j2ee":icon,
"http://java.sun.com/xml/ns/j2ee":distributable,
"http://java.sun.com/xml/ns/j2ee":context-param,
"http://java.sun.com/xml/ns/j2ee":filter,
"http://java.sun.com/xml/ns/j2ee":filter-mapping,
"http://java.sun.com/xml/ns/j2ee":listener,
"http://java.sun.com/xml/ns/j2ee":servlet,
"http://java.sun.com/xml/ns/j2ee":servlet-mapping,
"http://java.sun.com/xml/ns/j2ee":session-config,
"http://java.sun.com/xml/ns/j2ee":mime-mapping,
"http://java.sun.com/xml/ns/j2ee":welcome-file-list,
"http://java.sun.com/xml/ns/j2ee":error-page,
"http://java.sun.com/xml/ns/j2ee":jsp-config,
"http://java.sun.com/xml/ns/j2ee":security-constraint,
"http://java.sun.com/xml/ns/j2ee":login-config,
"http://java.sun.com/xml/ns/j2ee":security-role,
"http://java.sun.com/xml/ns/j2ee":env-entry,
"http://java.sun.com/xml/ns/j2ee":ejb-ref,
"http://java.sun.com/xml/ns/j2ee":ejb-local-ref,
"http://java.sun.com/xml/ns/j2ee":service-ref,
"http://java.sun.com/xml/ns/j2ee":resource-ref,
"http://java.sun.com/xml/ns/j2ee":resource-env-ref,
"http://java.sun.com/xml/ns/j2ee":message-destination-
ref, "http://java.sun.com/xml/ns/j2ee":message-
destination, "http://java.sun.com/xml/ns/j2ee":locale-
encoding-mapping-list}' is expected.
Column Number: + 8
Line Number: + 3
SAXParserException: org.xml.sax.SAXParseException:
```

```
cvc-datatype-valid.1.2.1: 'aa' is not a valid value for
'integer'.
Column Number: + 46
Line Number: + 7
SAXParserException: org.xml.sax.SAXParseException:
cvc-complex-type.2.2: Element 'load-on-startup' must
have no element [children], and the value must be valid.
Column Number: + 46
Line Number: + 7
2006-03-31 14:24:55.727 ERROR main [XmlDocument] An
error occurred during parsing.
2006-03-31 14:24:55.727 WARN main [DeployTool] Deploy
module failed due to exception:
com.sybase.djc.xml.XmlException: An error occurred
during parsing.
2006-03-31 14:24:55.742 ERROR main [DeployTool] Failed
to deploy module:
D:\xfsong\workhome\nbprojects\WebTest\dist\WebTest.war
2006-03-31 14:24:55.742 INFO main [DeployTool] Running
configuration script: webapp-webtest
D:\sybase\60069\bin\undeploy.bat webapp-webtest
2006-03-31 14:24:56.416 WARN main [UndeployTool]
Warning: No config file found for webapp-webtest
Warning: No config file found for webapp-webtest
2006-03-31 14:24:56.573 INFO main [UndeployTool]
Undeployed: webapp webtest
Undeployed: webapp webtest
```

Resource reference and mixed-component type failures

- If, for any reason, any component in a migrated package does not migrate successfully, none of the package is migrated. For example, if an EJB component has a resource reference to a connection cache that does not actually exist.
- If the components in in a package to be migrated are of different types, the package does not migrate. For example, a package that contains both EJB components and CORBA components does not migrate. See “Packages with mixed component types” on page 15 for more information.

In either of these situations, you see errors similar to these:

```
Warning: Component 'MyEjb/MyEjb ' is not migrateable
because it contains a resource-ref which links to an
unmigrateable ConnCache
```

```
Warning: Cannot migrate Application 'MyApp' because the
Package 'MyEjb' contains unmigrateable components or a
mix of component types.
```

Migrating Open Client/Server entries

EAServer 5.*x* contains a pre-installed version of Open Client/Server™, as well as an optional version located in the *%JAGUAR%\openclient* directory. You must configure either *%JAGUAR%\ini\sql.ini* or *%JAGUAR%\openclient\ini\sql.ini* to have an Open Client/Server entry. These entries are not migrated from EAServer 5.*x* to EAServer 6.0. You must manually migrate your Open Client/Server configurations to EAServer 6.0 by copying the required files to the new installation. See the Open Client/Server documentation set for more information.

Ensuring sample applications migrate correctly

Migrating the smarticket sample application

After migration, you must make these changes to the smarticket application:

1 Copy the smarticket folder (which contains *bin\smarticket.jar*) to the *$DJC_HOME/deploy/webapps/smarticket.web* directory.

2 Modify MIDlet-Jar-URL, SMARTicket-Servlet-URL, SMARTicket-Poster-URL, SMARTicket-Splash-URL in the *smarticket.jar* file as reflected in the smarticketapp configuration on EAServer 6.0.

Migrating the ART sample application

After migration, you must make these changesto the sample ART application:

1 Copy *$JAGUAR/bin/ART.properties* to the *$DJC_HOME* directory.

2 Verify that the ArtCache connection cache migrates from EAServer 5.*x* to EAServer 6.0.

3 After the EJB module nileejb and Web module nileejbweb are migrated to EAServer 6.0, set the nieejbweb namedclassloader's parent to nileejb namedclassloader.

EAServer 5.x users are not migrated to EAServer 6.0

Sybase recommends that you use the new roles defined in EAServer 6.0, and not jagadmin. The “Admin Role” role, of which jagadmin is a member is not automatically migrated. This may cause problems for migrated applications that use “Admin Role”.

When you migrate a Web application from EAServer 5.x to EAServer 6.0, J2EE roles and mapped EAServer roles defined in this Web application are migrated, but EAServer 5.*x* users are not. For example, the migrated sample Tier1 Web application does not have correct role name mapping, if the “Admin Role” from EAServer 5.x uses the default values.

In this case, the Web application in EAServer 5.x has the J2EE role “EAF Admin” mapped to the EAServer “Admin Role”. In the file *$DJC_HOME\config\webapp-eaf-user.xml*, you see this mapping:

```
<setProperties securityDomain="default">
<addRole role="eaf.EAF Admin" toRole="Admin Role"/>
</setProperties>
```

To work around this issue, modify *$DJC_HOME\config\webapp-eaf-user.xml*, to add the jagadmin user to the “Admin Role”, and run the configuration file.

Setting the context path and httpContexts property

After migrating a Web application to EAServer 6.0, you may have to set the context path to the application’s home page, and set the `httpContexts` property to “none.” For example, in the sybase.com application in EAServer 5.*x*, modify the EASDefault context path and set the sybase.com context path to “/”, so when you enter the URL http://<hostname>:8080/, the application’s home page appears.

Set the context path using *sybase-webapp-config.xml* in the EAServer 6.0 *WEB-INF* directory. For example:

```
<project name="webapp-sybase_com-user"
default="configure-user">
<import file="ant-config-tasks.xml"/>
<property name="web.contextPath" value="/"/>
</project>
```

For information about setting the server’s `httpContexts` property to “none, ” see Chapter 3, “Creating and Configuring Servers” in the *EAServer System Administration Guide*.

NoClassDef Found errors because stubs are not migrated

Stubs from EAServer 5.*x* are not migrated to EAServer 6.0. If you have components that make CORBA calls to other components, generate the stubs in EAServer 6.0. For example, you can successfully migrate a Web application and package of CORBA components used by the Web application, but the Web application generates NoClassDefFound errors because the stubs for the CORBA components have not been generated.

In EAServer 6.0 is to use jagtool to generate stubs. For example:

```
jagtool -n PORT -h HOSTNAME -u USERNAME -p PASSWORD gen_stubs -compilejavastubs
true -javastubtype CORBA Package:EAFExtension
```

Migration tool randomly fails to respond

occasionally, the migration tool may stop responding during either scan or migrate operations. You can tell if the tool has stopped responding if either:

- You stop seeing status messages, or
- CPU utilization goes down in the Task Manager.

To resolve this condition, perform one of the following:

- Kill the entire Java process that is running the migration tool, then re-run the migration with the djc.migratejagtoolTimeout value set:

  Set the Java system property djc.migrateJagtoolTimeout which specifies the maximum amount of time (in seconds) that the migration tool waits for a jagtool command to finish. Set the timeout value sufficiently high so that jagtool processes have enough time to complete a given task, 5 minutes. To launch the migration tool and set the property to 5 minutes, enter:

  ```
  migrate.bat -Ddjc.migrateJagtoolTimeout=600
  ```

- Terminate the jagtool Java process that is not responding.

  Once you have identified and terminated the offending process, migration can continue successfully, although you may see other jagtool calls that do not respond, requiring you to terminate those processes as well.

ResultSetMetaData implementation differences between JConnect drivers

There is a difference between jConnect 5 and jConnect 6 JDBC drivers in the behavior of ResultSetMetaData::getColumnLabel() and ResultSetMetaData::getColumnName():

- jConnect 5 – both APIs return the column label.
- jConnect 6 – getColumnLabel() returns the column label, while getColumnName() returns the actual name of the column.

If a client application uses getColumnName() in jConnect 5, you may need to change it to use getColumnLabel() instead. For example, let's say that an application in EAServer 5.*x* retrieves a column label by calling getColumnName(), then retrieves the data using ResultSet.getString(columnLabel). Because EAServer 6.0 uses jConnect 6 by default, this application now fails, because it is trying to since EAServer 6.0 uses JConnect 6 by default.

Here is a code sample to further illustrate this situation:

```
package test;
import java.sql.*;
public class TestJDBC {
       public static void main(String[] args) throws Exception{
//             Class.forName("com.sybase.jdbc2.jdbc.SybDriver");
               Class.forName("com.sybase.jdbc3.jdbc.SybDriver");
               Connection conn =
DriverManager.getConnection("jdbc:sybase:Tds:i18n280:4000/tck141db", "cts1",
"cts123");
             PreparedStatement ps = conn.prepareStatement("select my_name=name
from sysobjects");
               ResultSet rs = ps.executeQuery();

               System.out.println("Column One: " +
rs.getMetaData().getColumnName(1));
               System.out.println("Column One's label : " +
rs.getMetaData().getColumnLabel(1));

               ps.close();
               conn.close();
       }
}
```

CHAPTER 4

EAServer 6.0 Properties

Topic	Page
Migrated properties	41

Migrated properties

This section describes the various EAServer 5.*x* properties and their EAServer 6.0 equivalents, including:

Application properties

Application property names are prefixed with `com.sybase.jaguar.application`.

EAServer 5.x `com.sybase.jaguar.application.` properties	EAServer 6.0 equivalent	Description
`applicationclients`	Specified in the "modules" property of the application.	Specifies the comma-separated list of J2EE application clients installed in this application.

EAServer 5.x `com.sybase.jaguar.application.` properties	EAServer 6.0 equivalent	Description
`classloaderpolicy`	`parentFirst` property of the NamedClassLoader for the application.	Specifies how the custom class loader (version 2) resolves version conflicts when you specify the same class at multiple levels in the class loader hierarchy.
`description`	<description> element in *application.xml*.	Contains a text description of the application.
`DOMfactory`	Not applicable. (Use JAXP1.2 standard procedures to specify)	Specifies the class name for a custom DOM XML parser factory class.
`jagmgr.DOMFactoryChoice`	Not applicable. (Use JAXP1.2 standard procedures to specify)	Specifies the choice of DOM XML parser factory.
`jagmgr.SAXFactoryChoice`	Not applicable. (Use JAXP1.2 standard procedures to specify)	Specifies the choice of SAX XML parser factory.
`jagmgr.XSLTFactoryChoice`	Not applicable. (Use JAXP1.2 standard procedures to specify)	Specifies the choice of XSLT XML parser factory.
`java.classes`	Same. Supported for all entity types.	Specifies Java classes and JAR files to be loaded by the application's custom class loader.
`large-icon`	<large-icon> element in *application.xml*.	Specifies the name of an icon file to represent the Web application.
`name`	<display-name> element in *application.xml* or EAR file name.	Specifies the application name.
`packages`	Specified in the "modules" property of application.	Specifies the comma-separated list of packages installed in this application.
`SAXfactory`	Not applicable. (Use JAXP1.2 standard procedures to specify).	Specifies the class name for a custom SAX XML parser factory class.
`security.identities`	Not applicable.	Specifies the list of trusted identities used for incoming component invocations when propagating client credentials from another server.
`security.identity`	Not applicable.	Specifies the identity used for outgoing component invocations when propagating client credentials to another server.

EAServer 5.x com.sybase.jaguar.application. properties	EAServer 6.0 equivalent	Description
security-role.<j2ee-role>	Specified in the Web application's user *config.xml* file.	Specifies a mapping from a J2EE role name used in the application to a role defined in the EAServer repository.
security-roles	<security-role> element in the *application.xml* file.	Specifies logical J2EE role names used in the application.
small-icon	<small-icon> element in the *application.xml* file.	Specifies the name of an icon file to represent the Web application.
webapplications	Specified in the "modules" property of application.	Specifies the comma-separated list of Web applications installed in this application.
XSLTfactory	Not applicable. (Use JAXP1.2 standard to specify).	Specifies the class name for a custom XSLT XML parser factory class.

Application client properties

Application client property names are prefixed with `com.sybase.jaguar.applicationclient.`

EAServer 5.x com.sybase.jaguar.applicationclient. property	EAServer 6.0 equivalent	Description
description	<description> element in *application-client.xml*.	Contains a text description of the application client.
ejb-ref	<ejb-ref> element in *application-client.xml*.	Specifies a list of EJB references that define aliased JNDI names for EJB components invoked by the application client.
env-entry	<env-entry> element in *application-client.xml*.	Environment properties allow you to specify global read-only data for use by the application client.
files	Not applicable.	Specifies additional files and Java classes to be distributed with the application client.
main-class-name	Not needed.	Specifies the main Java class of the application client.
name	Not needed. Derived from config file name.	Specifies the application client name.

EAServer 5.x com.sybase.jaguar.applicationclient. property	EAServer 6.0 equivalent	Description
`resource-env-ref`	<resource-env-ref> element in *application-client.xml*.	Resource environment references are logical names applied to objects administered by EAServer
`resource-ref`	<resource-ref> element in *application-client.xml*.	Specifies aliased JNDI names for database connections, JavaMail sessions, and URL factories used by the application client.

Component properties

Component property names are prefixed with `com.sybase.jaguar.component.`

Note For CORBA Java, C++, and PowerBuilder component migration, component property files are copied directly from exported Sybase JAR files to EAServer 6.0 repository without modification.

EAServer 5.x `com.sybase.jaguar.component.` property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
`auto.failover`	<automaticFailover> property.	Yes.	Specifies whether client proxies can transparently fail over to a different server.
`auto.profiles`	Not applicable.	No.	Obsolete.
`bind.naming`	Not needed for EJB/MDB.	Not needed.	Specifies the name with which the component is bound to the name service. If not specified, the default is package/component where package is the EAServer package name, and component is the component name.
`bind.object`	Not needed for EJB/MDB.	Yes.	Specifies whether instances are bound to client's object reference.
`bind.thread`	Not needed for EJB/MDB.	Yes.	Specifies whether instances are bound to the thread that created them.

EAServer 5.x `com.sybase.jaguar.component.` property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
`classloaderpolicy`	<parentFirst> property for EJB's ClassLoader.		Specifies how the custom class loader (version 2) resolves version conflicts when you specify the same class at multiple levels in the class loader hierarchy.
`cmp_iso_level`	"isolationLevel" attribute of <persistentObject> property.	Not needed.	Specifies the effective transaction isolation level for EJB CMP entity beans.
`cmp.version`	<cmp-version> in *ejb-jar.xml*.	Not needed.	For EJB 2.0 entity beans that use CMP (automatic persistence), specifies the CMP model version. (1.1 default or 2.0).
`code.set`	Not needed for EJB/MDB.		For PowerBuilder, C, or C++ components, specifies the coded character set name used to encode character and string parameter data.
`com.progid`	Not needed for EJB/MDB.	COM components not supported.	Specifies the progid that the component uses in the COM Automation Server Registry.
`context`	Not needed for EJB/MDB.		Specifies the name of the component's context IDL interface: for example, `CtsComponents::ObjectContext`.
`control`	Not needed for EJB/MDB.	Yes.	Specifies the name of the component's control IDL interface, for example `CtsComponents::ObjectControl`.
`cpp.class`	Not needed for EJB/MDB.	Yes.	For a C++ component, specifies the implementation class name.
`cpp.copy`	Not needed for EJB/MDB.	Yes.	For C and C++ components, specifies whether the server should copy the component library before running it.
`cpp.debug`	Not needed for EJB/MDB.	Yes.	For a C++ component, specifies whether to catch exceptions.

EAServer 5.x com.sybase.jaguar.component. property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
cpp.library	Not needed for EJB/MDB.	Yes.	For C and C++ components, specifies the name of the DLL or shared library that contains the implementation class.
cpp.process	Not needed for EJB/MDB.	Yes.	For C++ components, specifies the name of an external process to run the component. If the property is not set, the component executes within the EAServer process.
cs.create	Not needed for EJB/MDB.	C components not supported.	For a C component, specifies whether the implementation has a create routine.
cs.destroy	Not needed for EJB/MDB.	C components not supported.	For a C component, specifies whether the implementation has a destroy routine
db.sequence			For an EJB CMP entity bean, or components of other types that use automatic persistence, specifies the database sequence name, if required by the database
debug			Specifies whether the server logs trace information for instance life cycle events such as creation, destruction, pooling, and so forth.
defer		Not needed.	In an EJB CMP entity bean that uses a Sybase CMP wrapper driver, specifies which SQL commands may be deferred to the end of the transaction.
destroyPooledInstancesOnShutdown			Overrides the com.sybase.jaguar.server.destroyPooledInstancesOnShutdown server property.
destroyPooledInstancesOnShutdownTimeout			How long to wait for each instance destruction method to return when destroying pooled instances during server shutdown

EAServer 5.x `com.sybase.jaguar.component.` property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
`dn.triggers`		Not needed.	For EJB CMP entity beans that use instance and query caching with database change notification enabled, this property enables automatic creation of database triggers to notify the EAServer cache manager when the table data changes.
`DOMfactory`	Not applicable. (Use standard JAXP procedures to specify).	Not needed.	Specifies the class name for a custom DOM XML parser factory class.
`ejb.home`	<home> in *ejb-jar.xml*.	Not needed.	For EJB components, specifies the home interface class name. Derived from `com.sybase.jaguar.component.home`
`ejb.key`	<prim-key-class> in *ejb-jar.xml*.	Not needed.	For an EJB entity bean component, specifies the class name for the primary key type. Derived from `com.sybase.jaguar.component.key`.
`ejb.local`	<local> in *ejb-jar.xml*.	Not needed.	For EJB 2.0 components, specifies the Java local interface name.
`ejb.local.home`	<local-home> in *ejb-jar.xml*.	Not needed.	For EJB 2.0 components, specifies the Java local home interface name.
`ejb.remote`	<remote> in *ejb-jar.xml*.	Not needed.	Specifies the remote interface class name for an EJB component.
`ejb-local-ref`	<ejb-local-ref> in *ejb-jar.xml*.	Not needed.	For EJB components, specifies a list of EJB local references that define aliased JNDI names for local components invoked by this component.
`ejb-ref`	<ejb-ref> in *ejb-jar.xml*.	Not needed.	For EBJ components, specifies a list of EJB references that define aliased JNDI names for components invoked by this component.
`env-entry`	<env-entry> in *ejb-jar.xml*.	Not needed.	For EJB components (EJB version 1.1 or later), specifies environment properties.

EAServer 5.x com.sybase.jaguar.component. property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
external.request.timeout			For components that run in an external server, specifies how long to wait for a response from the external server before returning an error to the client.
external.servername			For stateless components, specifies the server name for external component execution.
external.serverstart.timeout			If the component is configured to run in an external server, specifies how long to wait for a response from the external server before returning an error to the client.
files	Not needed for EJB/MDB.	Yes.	Specifies additional files that are included when the component is archived with the package export feature or replicated with the synchronize feature.
files.corbastubs	Not needed for EJB/MDB.	Not needed.	Specifies the files that implement the component's Java/CORBA stubs.
files.ejbstubs	Not needed for EJB/MDB.	Not needed.	Specifies the files that implement the component's EJB stubs.
generateKey	Not needed for EJB/MDB.		For an EJB CMP entity bean, or other types of components that use automatic persistence, specifies whether the key values the mapped database table are automatically generated.
home	Not needed for EJB/MDB.	Not needed.	Specifies the component's home IDL interface. EJB components must have a home interface, and other component types must have one to support access from EJB clients
home.ids	Not needed for EJB/MDB.		Specifies repository IDs for the component's home IDL interface. Derived from com.sybase.jaguar.component.home.

EAServer 5.x com.sybase.jaguar.component. property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
identitytype	Not needed for EJB/MDB.		Specifies whether to use the connection user name or SSL certificate for role-based authorization.
ids	Not needed for EJB/MDB.		Specifies repository IDs for the component's IDL interfaces.
instancePool	Not needed for EJB/MDB.	Yes.	Constrains the component to run in the specified instance pool (system or bindThread).
interfaces	Not needed for EJB/MDB.	Yes.	Specifies the IDL interfaces that the component supports for client use.
iso_level	Not supported.	Not needed.	For EJB 1.0 components, specifies the isolation level for transactions begun by the component's methods.
java.class	<ejb-class> in *ejb-jar.xml*.	Yes.	For Java components (CORBA and EJB), specifies the name of the Java implementation class.
java.classes	Not needed.	Not needed.	For Java components (CORBA and EJB), specifies additional classes to be loaded by the component class loader.
key	Not needed for EJB/MDB.	Yes.	For an entity component, specifies the IDL datatype for the primary key.
key.tc	Not needed for EJB/MDB.		Specifies the type code string for the primary key IDL type.
keys	Not needed for EJB/MDB.	Yes.	For entity components, specifies the name of an IDL type definition (typedef) for a sequence of the component's primary key datatype. This type is used when generating the skeleton and implementation classes for the component.
list	Not needed for EJB/MDB.		For an EJB 1.0 entity bean, specifies the IDL datatype for a sequence of the remote interface type. In an entity bean's IDL home interface, methods that return a collection of instances must return the type name specified by this property.

EAServer 5.x com.sybase.jaguar.component. property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
listener	Not needed for EJB/MDB.	Not needed.	For message-driven beans (MDBs), specifies the message listener package and component.
load	Not needed for EJB/MDB.	Not supported.	Obsolete.
local	Not needed for EJB/MDB.	Not needed.	For EJB 2.0 components, specifies the name of the IDL local interface.
local.home	Not needed for EJB/MDB.	Not needed.	For EJB 2.0 components, specifies the name of the IDL local home interface.
lwc	Not supported.	Not needed.	For EJB components only. Enables the EAServer lightweight container (LWC) for calls to this component from EJBs or servlets and JSPs hosted in the same server.
lwc.enableSkeletons	Not supported.	Not needed.	Enables LWC calls to EJB components from the servlets and JSPs hosted in the same server. Such calls are not supported unless this option is set.
maxpool	Not supported.		When instance pooling is enabled, specifies the maximum pool size.
maxwait	Not supported.		This setting applies only when the com.sybase.jaguar.component.objects property is set to specify a limit on the number of simultaneous active instances
mdb.acknowledge-mode	<acknowledge-mode> in *ejb-jar.xml*.	Not needed.	Specifies the acknowledgment mode for MDBs that manage their own transactions.
mdb.destination-type	<destination-type> in *ejb-jar.xml*.	Not needed.	Specifies whether the MDB is associated with a JMS topic or message queue
mdb.message-selector	<message-selector> in *ejb-jar.xml*.	Not needed.	For an MDB associated with a message queue, specifies the message selector for a message queue. The message service uses the selector to filter the message that it delivers to the queue.

EAServer 5.x `com.sybase.jaguar.component.` property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
`mdb.subscription-durability`	<subscription-durability> in *ejb-jar.xml*.	Not needed.	When the MDB is associated with a topic, specifies whether the topic is durable or nondurable.
`methods`	Not applicable.	Yes.	Used to store the component's nondefault method properties.
`minpool`			When instance pooling is enabled, specifies the minimum pool size.
`model`	Not needed.		Specifies the component model (ejb, cts, mts).
`model.version`			For an EJB component, specifies the EJB specification version.
`monitor`		Yes.	Assigns this component to a thread monitor.
`monitor.MaxRespTime`		Yes.	Specifies the maximum allowable average response time for the component. If the average method completion time rises above this limit, EAServer blocks creation of additional instances of this component until the average drops below the specified limit.
`monitor.MinInstance`		Yes.	When response time monitoring is in effect (`com.sybase.jaguar.component.monitor.MaxRespTime` is set to a non-negative number), specifies the minimum number of instances that must be allowed to execute regardless of observed response times.
`name`	Not needed.	Yes.	Specifies the name of the component and the package in which it is installed.
`objectCache`			Specifies the name of the component that implements object caching for entity components that use automatic persistence and EJB CMP entity beans.

EAServer 5.x com.sybase.jaguar.component. property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
objectCache.sizeCheckInterval		Not needed.	For EJB CMP entity beans that use the object cache, specifies how often EAServer checks the size of the entry before placing it in the cache.
objects		No.	Specifies the maximum number of component instances that can exist at once. For a C++ component that runs as an external process, specifies the maximum number of simultaneously running external processes.
passByReference		Not needed.	Enables the proprietary EJB pass-by-reference in-server invocation mechanism supported by some other J2EE vendors.
pb.appname	Not for EJB/MDB.		For PowerBuilder components, the name of the PowerBuilder application that contains the NVO that implements the component.
pb.class	Not for EJB/MDB.	Yes.	For PowerBuilder components, specifies the name of the nonvisual object that implements the component's methods.
pb.cookie	Not for EJB/MDB.	Yes.	For versioned PowerBuilder components, contains version cookie data.
pb.debug	Not for EJB/MDB.		For PowerBuilder components, specifies whether the component can be debugged while executing.
pb.librarylist	Not for EJB/MDB.	Yes.	For PowerBuilder components, specifies library files that are required to run the object.
pb.live_edit	Not for EJB/MDB.		For PowerBuilder components, enables or disables the live editing feature.
pb.version	Not for EJB/MDB.	Only PB 10 and above are Supported.	For PowerBuilder components, specifies the required version of the PowerBuilder virtual machine.
pooling		Yes.	Specifies whether component instances are pooled for reuse by multiple client sessions.

EAServer 5.x `com.sybase.jaguar.component.` property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
`ps`		Yes.	For an entity or stateful session component, specifies how persistence is performed.
`ps.class`	Not needed.	Yes.	For entity components, specifies the generated class to handle component persistence when `com.sybase.jaguar.component.ps` property is "generated".
`qop`	Not for EJB/MDB.		Minimum quality of protection required to access component.
`queue`		Not needed.	For MDBs associated with a message queue, specifies the name of the message queue.
`reentrant`	<reentrant> in *ejb-jar.xml*		For entity components, including EJB entity beans, specifies whether a component can recursively call itself, or participate in an intercomponent call sequence that creates recursion.
`refresh`	Always refreshable.	Always refreshable.	Specifies whether the component can be refreshed, that is, whether a new implementation can be loaded while the server is running
`remote`	Not needed.	Not needed.	Specifies the name of the remote (main) IDL interface. EJB components must have a remote interface, and other component types must have one to support access from EJB clients.
`resource-env-ref`	<resource-env-ref> in *ejb-jar.xml*.	Not needed.	Resource environment references are logical names applied to objects administered by EAServer.
`resource-ref`	<resource-ref> in *ejb-jar.xml*.	Not needed.	For EJB components, specifies aliased JNDI names for database connections, JavaMail sessions, and URL factories used by the component.
`retry.timeout`			For an EJB MDB component, specifies how long the server should redeliver a message after the MDB has thrown an exception while processing the message.

EAServer 5.x `com.sybase.jaguar.component.` property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
`roles`	Not for EJB/MDB.	Yes.	For component types other than EJB, specifies roles that a user must belong to invoke the component.
`runasidentity`		Not needed.	For EJB 1.0 components, specifies an identity name used for intercomponent calls if the `com.sybase.jaguar.component.runasmode` property is "specified".
`runasmode`		Not needed.	For EJB 1.0 components, specifies the user identity that is assumed for intercomponent calls.
`SAXfactory`	Not applicable. (Use standard JAXP procedures to specify).	Not needed.	Specifies the class name for a custom SAX XML parser factory class.
`security.runasidentity`	<run-as> in *ejb-jar.xml*.	Not needed.	For EJB 2.0 components, specifies the run-as identity used for intercomponent calls. If this property is not set, intercomponent calls use the client identity.
`security-role-ref.<j2ee-role-ref>`	<security-role-ref> in *ejb-jar.xml*.	Not needed.	For EJB components, maps a role reference specified in the `com.sybase.jaguar.component.security-role-refs` property to a J2EE role name defined in package or application properties.
`security-role-refs`	<security-role-ref> elements in *ejb-jar.xml*.	Not needed.	For EJB components, specifies role reference names used within `isCallerInRole` method invocations in the component implementation.
`selectForUpdate`		No.	For components that use automatic persistence, such as EJB CMP entity beans, specifies whether queries create exclusive locks on selected rows
`selectWithLock`		No.	For components that use automatic persistence, such as EJB CMP entity beans, specifies whether queries create shared locks on selected rows.

EAServer 5.x `com.sybase.jaguar.component.` property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
`sharing`		Yes.	Specifies whether multiple component instances can be created. A shared component serves all client requests with one instance.
`softLock`			For EJB CMP entity beans that use an isolation level of `repeatable_read_with_cache`, enables soft locking.
`softLock.timeout`			When soft locking is enabled, specifies the timeout period for soft-locked rows.
`state`		Yes.	Specifies the datatype for component state information. Applies to entity components or stateful components that use automatic storage other than EJB entity beans and stateful session beans.
`state.gs`		Yes.	For entity components or stateful components that use automatic persistence, specifies component methods that set and retrieve component state.
`stateless`	<session-type> in *ejb-jar.xml*.	Yes.	For EJB session beans and non-EJB components that use the control interface `CtsComponents::ObjectControl`, specifies whether the component is stateless, that is, whether an instance should be bound to a client session for its lifetime.
`storage`	<persistentObject> property	Yes.	For entity or stateful session components that use automatic persistence or failover support, specifies the storage component.
`store`			Tells the server when to call ejbStore.
`sync`			Specifies whether the component is included in the files replicated by a synchronization operation.

EAServer 5.x com.sybase.jaguar.component. property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
thread.safe	Not for EJB/MDB	Yes.	Specifies whether multiple component instances can execute concurrently, or whether a shared component can execute simultaneously on multiple threads. If this option is disabled, the server serializes all invocations of component methods.
timeout		Yes.	Specifies how long, in seconds, an active component instance can remain idle between method calls before the client's proxy becomes invalid.
timestamp			For entity components that use automatic persistence with mapped table fields, specifies how the server uses optimistic concurrency control to prevent overlapping updates to the same column.
tlc.sort		Not needed.	For EJB CMP entity beans, specifies that transaction local cache entries for this component should be sorted before ejbStore is called.
topic		Not needed.	For MDBs associated with a message topic, specifies the name of the topic.
touchColumn		No.	Used when the com.sybase.jaguar.component.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.
trace	<tracePublicMethods> property.		Enables additional debug tracing for the component, such as logging method names, client IP addresses, and user names for each invocation of a business method.

EAServer 5.x com.sybase.jaguar.component. property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
transient			Applies to stateful components only. Specifies whether instances can be run on multiple servers in a cluster or survive a server restart.
ts.length			When com.sybase.jaguar.component.timestamp specifies a column name, this property specifies the datatype and length.
ts.triggers		Not needed.	For EJB CMP entity beans that use a table-level timestamp, specifies whether EAServer must create triggers to update the timestamp when a row in the mapped table is inserted, modified, or deleted.
tx_control	Not needed.		Specifies whether control interface methods execute in the context of a server-managed transaction.
tx_outcome	Not needed.		Determines whether a CORBA::TRANSACTION_ROLLEDBACK exception is thrown to the client when a transaction is rolled back.
tx_retry	"retry" attribute of <transaction> property.	Not needed.	For EJB CMP entity beans, determines whether EAServer automatically replays transactions that fail.
tx_timeout			Specifies the maximum duration of an EAServer transaction begun by the component.
tx_type	<transaction-type> and <tran-attribute> in *ejb-jar.xml*.	Yes.	Specifies how methods in your component participate in transactions.
tx_vote		Yes.	Specifies whether the component is automatically deactivated after each method invocation, or deactivated at explicit transaction boundaries. For component types to which it applies, setting this property to true effectively marks the component as stateless.

EAServer 5.x `com.sybase.jaguar.component.` property	EAServer 6.0 EJB/MDB equivalent	CORBA component support?	Description
`type`	Not needed, derived from *ejb-jar.xml*.	Yes.	Specifies the component type.
`XSLTfactory`	Not applicable. (Use standard JAXP procedures to specify).	Not needed.	Specifies an XSLT Parser factory class name.
`description`	<description> in *ejb-jar.xml*.	Yes.	Specifies a text description of the component.

Connector properties

Connector property names are prefixed with `com.sybase.jaguar.connector`.

EAServer 5.x `com.sybase.jaguar.connector.` property	EAServer 6.0 equivalent	Description
`auth-mechanism`	<authentication-mechanism-type> in *ra.xml*.	Specifies the authentication mechanism supported by the resource adapter (connector).
`config-property`	<config-property> elements in *ra.xml*.	Defines configuration properties for a managed connection factory.
`connection-impl-class`	<connection-impl-class> in *ra.xml*.	Specifies the connector's implementation class.
`connection-interface`	<connection-interface> in *ra.xml*.	Specifies the name of the connection interface supported by the connector.
`connectionfactory-impl-class`	<connectionfactory-impl-class> in *ra.xml*.	Specifies the connection factory class that implements the connector-specific ConnectionFactory interface.
`connectionfactory-interface`	<connectionfactory-interface> in *ra.xml*.	Specifies the connection factory interface supported by the connector.
`display-name`	<display-name> in *ra.xml*.	The name that identifies this connector in EAServer Manager.
`eis-type`	<eis-type> in *ra.xml*.	Specifies the Enterprise Information System (EIS) type to which connections are made.
`enabled`	Always enabled.	Specifies whether the connector is enabled or disabled.

EAServer 5.x `com.sybase.jaguar.connector.` property	EAServer 6.0 equivalent	Description
`icon.large-icon`	<large-icon> in *ra.xml*.	Specifies the name of the large icon file associated with the connector.
`icon.small-icon`	<small-icon> in *ra.xml*.	Specifies the name of the small icon file associated with the connector.
`idletimeout`	Not applicable.	Specifies the number of seconds an idle connection remains in the pool before it is dropped.
`java.classes`	Not applicable.	Specifies additional classes to be reloaded when the connector is refreshed.
`managedconnectionfactory-class`	<managedconnectionfactory-class> in *ra.xml*.	Specifies the Java class that implements the javax.resource.spi.ManagedConnectionFactory interface.
`name`	Derived from the connector RAR file.	The name that identifies this connector in the repository.
`queuesize`	Not applicable.	Specifies the connector's connection pool size.
`reauthenticationsupport`	<reauthentication-support> in *ra.xml*.	Specifies whether the connector supports reauthentication of an existing managed connection factory instance.
`security-permission`	<security-permission> elements in *ra.xml*.	Where security-permission is other than those required by the default permission set defined in the connector specification.
`spec-version`	<spec-version> in *ra.xml*.	Specifies the connector architecture specification version number that is supported by the connector.
`transaction-support`	<transaction-support> in *ra.xml*.	Specifies the level of transaction support provided by the connector.
`vendor-name`	<vendor-name> in *ra.xml*.	Specifies the connector's vendor name.
`version`	<version> in *ra.xml*.	Specifies the version number of the connector software.

Filter properties

Filter property names are prefixed with `com.sybase.jaguar.filter`.

EAServer 5.x `com.sybase.jaguar.filter.` property	EAServer 6.0 equivalent	Description
`class`	<filter-class> in *web.xml*.	The Java class that implements the filter.
`description`	<description> in *web.xml*.	An optional text description of the filter.
`init-param`	<init-param> in *web.xml*.	Initialization parameters for the filter.
`large-icon`	<large-icon> in *web.xml*.	Specifies the name of the large icon file associated with the filter.
`name`	<filter-name> in *web.xml*.	The name that identifies this filter in the repository.
`small-icon`	<small-icon> in *web.xml*.	Specifies the name of the small icon file associated with the filter.

Package properties

Package property names are prefixed with `com.sybase.jaguar.package`.

EAServer 5.x `com.sybase.jaguar.package.` property	EAServer 6.0 EJB/message driven bean (MDB) equivalent	CORBA Component support?	Description
`application`	Not needed.	Not needed.	Specifies the application that this package is installed in, if any.
`classloaderpolicy`	parentFirst property of the package's ClassLoader.	Yes.	Specifies how the custom class loader (version 2) resolves version conflicts when you specify the same class at multiple levels in the class loader hierarchy.
`code.set`	Not needed.		Specifies the default for the `com.sybase.jaguar.component.code.set` component property.
`DOMfactory`	Not applicable. (Use JAXP1.2 standard procedures to specify).	Not needed.	Specifies the class name for a custom DOM XML parser factory class.
`files`	Not needed.	Not needed.	Specifies files to be included when the package is exported to an EAServer JAR archive or replicated to another installation using synchronization.

EAServer 5.x `com.sybase.jaguar.package.` property	EAServer 6.0 EJB/message driven bean (MDB) equivalent	CORBA Component support?	Description
`files.corbastubs`	Not needed.	Not needed.	Specifies the files that implement Java/CORBA stubs for the components in the package.
`files.ejbstubs`	Not needed.	Not needed.	Specifies the files that implement EJB stubs for the components in the package.
`java.classes`	Not needed.	Not needed.	Specifies Java classes and JAR files to be loaded by the package's custom class loader.
`name`	Not needed.	Yes.	Specifies the package name.
`roles`	Not needed.		Specifies roles that a user must belong to in order to invoke the components (other than EJB components) installed in this package.
`runasidentity.<id>`	Not applicable. Use EJBs run-as.	Not needed.	Maps an identity name used in EJB (1.0 and 2.0) component configuration to an identity defined in the EAServer repository.
`SAXfactory`	Not applicable. (Use JAXP1.2 standard procedures to specify).	Not needed.	Specifies the class name for a custom SAX XML parser factory class.
`security-role.<j2ee-role>`	<addRole> elements in user config file.	Not needed.	Specifies a mapping from a J2EE role name used in the package to a role defined in the EAServer repository.
`security-roles`	<security-role> elements in *ejb-jar.xml*.	Not needed.	Specifies logical J2EE role names used in the package.
`XSLTfactory`	Not applicable. (Use JAXP1.2 standard procedures to specify).	Not needed.	Specifies the class name for a custom XSLT XML parser factory class.
`schema:<schema-name>`	Not applicable.	Not needed.	Specifies the EJB 2.0 (or later) component that represents a schema name used in EJB-QL queries.

Servlet properties

Servlet property names are prefixed with `com.sybase.jaguar.servlet`.

EAServer 5.x `com.sybase.jaguar.servlet.` property	EAServer 6.0 equivalent	Description
`cache`	Servlet/JSP's "cacheResponse" attribute tag.	Enables and disables servlet response caching.
`cache.entire-tree`	Not applicable.	Specifies whether to cache all the pages or files that are invoked by a Web component.
`cache.locale-sensitive`	"localeSensitive" property of "cacheResponse" tag. **Note** Default changed from false in EAServer 5.*x* to true in EAServer 6.0.	Specifies whether to cache all the pages or files that are invoked by a Web component.
`cache.message-topics`	Not applicable.	When response caching is enabled, specifies message service topic names used to synchronize the cache with an external storage mechanism such as a database.
`cache.request-headers`	"requestHeaders" property of "cacheResponse" tag.	When response caching is enabled, specifies request headers to include in the cache key.
`cache.request-parameters`	"requestParameters" property of "cacheResponse" tag.	When response caching is enabled, specifies request parameters to include in the cache key.
`cache.session-attributes`	"sessionAttributes" property of "cacheResponse" tag.	When response caching is enabled, specifies session attributes to include in the cache key.
`cache.timeout`	"timeOut" property of "cacheResponse" tag.	When response caching is enabled, specifies the cache timeout value.
`cache.use-sessionid`	sessionLocal property of "cacheResponse" tag.	When response caching is enabled, specifies whether to include the session ID in the cache key.
`description`	Not applicable.	An optional description of the servlet or JSP.
`destroy.wait-time`	Not applicable.	Overrides the server or Web application servlet destroy timeout setting for this servlet.

EAServer 5.x `com.sybase.jaguar.servlet.` property	EAServer 6.0 equivalent	Description
`files`	Not applicable.	Specifies additional files to be included if this servlet is archived or replicated to another installation using the synchronize feature
`init-param`	<init-param> in *web.xml*.	Specifies initialization parameters for the servlet.
`init.timeout`	Not applicable.	Specifies how long the server should wait for the servlet's init method to return.
`javacache.enabled`	Not applicable.	Enables the servlet Java cache mechanism for this servlet or JSP.
`javacache.maxsize`	Not applicable.	When using the Java cache mechanism, specifies the size, in kilobytes, of the largest reply that can be cached. Responses larger than this are not cached.
`javacache.session`	Not applicable.	When using the Java cache mechanism, specifies how session cookie settings are treated in response headers when using the Java cache mechanism.
`javacache.timeout`	Not applicable.	When using the Java cache mechanism, specifies the time, in seconds, that cached responses remain valid.
`java.class`	<servlet-class> in *web.xml*.	Specifies the servlet implementation class.
`java.classes`	Not applicable.	Specifies additional classes to be reloaded when the servlet is refreshed.
`jsp.compile-extra-cp`	Not applicable.	Specifies additional JAR files and directories to include in the JSP compiler class path
`jsp.compile-use-eas-cp`	Not applicable.	Applies to JSPs only. Specifies whether the EAServer process CLASSPATH should be included in the compilation class path when compiling this JSP.
`jsp.compile-use-third-party`	Not applicable.	Applies to JSPs only. Specifies whether JAR files in the EAServer *java/lib* directory are automatically included in the class path when compiling this JSP.
`jsp-file`	<jsp-file> in *web.xml*.	If this servlet is generated from a JSP file, specifies the *.jsp* file name.
`large-icon`	<large-icon> in *web.xml*.	Specifies the name of the large icon file associated with the servlet.
`load-on-startup`	<load-on-startup> in *web.xml*.	Specifies whether the servlet is loaded at server start-up time, or after the first client request.

EAServer 5.x `com.sybase.jaguar.servlet.` property	EAServer 6.0 equivalent	Description
`name`	<servlet-name> in *web.xml*.	The servlet name.
`security.runasidentity`	<run-as> in *web.xml*.	Specifies the run-as identity used for component calls. If this property is not set, intercomponent calls use the client identity
`servletorjsp`	Not applicable.	Specifies whether this servlet is generated from a JSP.
`session.allowed`	Not applicable.	For servlets not in a Web application, specifies whether the servlet can use sessions.
`session.timeout`	<session-timeout> in *web.xml*.	Specifies the session timeout.
`singlethread`	Not applicable.	Specifies whether an instance of the servlet class can be run simultaneously on multiple threads
`singlethread.poolsize`	Not applicable.	If the servlet is single-threaded, specifies the number of threads to run the servlet on. More threads may decrease the average client response time by eliminating the need to serialize requests.
`small-icon`	<small-icon> in *web.xml*.	Specifies the name of the small icon file associated with the servlet.

Web application properties

Web application property names are prefixed with `com.sybase.jaguar.webapplication`.

EAServer 5.x `com.sybase.jaguar.webapplication.` property	EAServer 6.0 equivalent	Description
`application`	Not needed (derived from property file name).	The name of the application to which this Web application belongs.
`locale-sensitive`	Not applicable. Uses the servlet-level cache property.	For installed servlets, specifies the default for the servlet property `com.sybase.jaguar.servlet.cache.locale-sensitive`.

EAServer 5.x `com.sybase.jaguar.webapplication.` property	EAServer 6.0 equivalent	Description
`cache.message-topics`	Not applicable.	For installed servlets, specifies the default for the servlet property `com.sybase.jaguar.servlet.cache.message-topics`.
`cache.request-headers`	Not applicable. Uses servlet-level cache property.	For installed servlets, specifies the default for the servlet property `com.sybase.jaguar.servlet.cache.request-headers`.
`cache.request-parameters`	Not applicable. Uses servlet-level cache property.	For installed servlets, specifies the default for the servlet property `com.sybase.jaguar.servlet.cache.request-parameters`.
`cache.session-attributes`	Not applicable. Uses servlet-level cache property.	For installed servlets, specifies the default for the servlet property `com.sybase.jaguar.servlet.cache.session-attributes`.
`cache.timeout`	Not applicable. Uses servlet-level cache property.	For installed servlets, specifies the default for the servlet property `com.sybase.jaguar.servlet.cache.timeout`.
`cache.use-sessionid`	Not applicable. Uses servlet-level cache property.	For installed servlets, specifies the default for the servlet property `com.sybase.jaguar.servlet.cache.use-sessionid`.
`charset.inputdata`	Not applicable.	Specifies the character set for JSP or servlet request body data.
`charset.inputparam`	Not applicable.	Specifies the character set for servlet and JSP request parameters.
`charset.jspcompile`	Not applicable.	Specifies the character set for JSP compilation.
`classloaderpolicy`	parentFirst property of the Web application's ClassLoader.	Specifies how the custom class loader (version 2) resolves version conflicts when you specify the same class at multiple levels in the class loader hierarchy.
`context-param`	<context-param> in *web.xml*.	Specifies the context initialization parameters for the servlets in the Web application.
`context-path`	“contextPath” property	The request-path prefix that clients use in URLs to access your Web application's static content, servlets, and JSPs.
`cookie.persistent`	Not applicable.	Specifies whether session data cookies are persistent or temporary

EAServer 5.x `com.sybase.jaguar.webapplication.` property	EAServer 6.0 equivalent	Description
`dependencies`	Not applicable.	Specifies dependencies on standard Java extensions.
`default.protectedpage`	Not applicable.	Specifies a default page to redirect to if a servlet uses direct form login to authenticate the user.
`destroy-wait-time`	Not applicable.	The number of seconds that EAServer should wait for servlet service calls to return before calling the servlet destroy method. Affects all servlets installed in the Web application.
`distributable`	<distributable> in *web.xml*.	Specifies whether multiple instances of the Web application can run in a distributed server environment on different servers.
`distribute.type`	Not applicable. (distributed session could only be persisted on database).	Specifies how EAServer replicates HTTP session data for distributable Web applications.
`DOMfactory`	Not applicable. (Use JAXP1.2 standard procedures to specify).	XSLT Parser factory class name.
`ejb-local-ref`	<ejb-local-ref> in *web.xml* and *bind:java:comp/env/ejb/.* .. in properties.	Specifies a list of EJB local references that define aliased JNDI names for local EJB components invoked by servlets in the Web application.
`ejb-ref`	<ejb-ref> in *web.xml* and *bind:java:comp/env/ejb/.* .. in properties.	Specifies a list of EJB references that define aliased JNDI names for EJB components invoked by servlets in the Web application.
`env-entry`	<env-entry> in *web.xml*.	Environment properties allow you to specify global read-only data for use by the servlets in the Web application. Servlets must use JNDI to retrieve environment properties, using the prefix java:comp/env in JNDI lookups.
`files`	Not applicable.	Specifies additional files to be included when the Web application is exported into a Jaguar JAR file or replicated using the synchronization feature.
`filter-mapping`	<filter-mapping> in *web.xml*.	Associates filters with servlets and URL paths.
`filters`	<filter> in *web.xml*.	Specifies filters installed in this Web application.

EAServer 5.x `com.sybase.jaguar.webapplication.` property	EAServer 6.0 equivalent	Description
`get-serverinfo-from`	Not applicable.	When using a Web server redirector, configures the source for information returned by the HTTPServletRequest methods getScheme, getServerPort, and getServerName.
`httpdomain.override`	Not applicable.	If set to true, the server HTTP Domain Name property `com.sybase.jaguar.server.http.domainname` is ignored for this Web application.
`init-timeout`	Not applicable.	Specifies how long to wait for each installed servlet's init method to return.
`jagmgr.DOMFactoryChoice`	Not applicable. (Use JAXP1.2 standard procedures to specify).	DOM Parser factory class name.
`jagmgr.SAXFactoryChoice`	Not applicable. (Use JAXP1.2 standard procedures to specify).	SAX Parser factory class name.
`jagmgr.XSLTFactoryChoice`	Not applicable. (Use JAXP1.2 standard procedures to specify)	XSLT Parser factory class name.
`jarlist`	Not applicable.	Specifies the class loading order when classes are loaded from JAR files in the *WEB-INF/lib* directory under the Web application's context root.
`java.classes`	Not applicable.	Specifies additional classes and JAR files to be loaded by the Web application's custom class loader, in addition to those deployed in the *WEB-INF/lib* and *WEB-INF/classes* directories.
`compile-extra-cp`	Not applicable.	Specifies additional JAR files and directories to include in the JSP compiler class path.
`compile-use-eas-cp`	Not applicable.	Specifies whether the EAServer process CLASSPATH should be included in the compilation class path when compiling JSPs.
`compile-use-thrid-party`	Not applicable.	Specifies whether JAR files in the EAServer *java/lib* directory are automatically be included in the class path when compiling JSPs.
`jspc-interval`	Not applicable.	Determines if and when the JSP runtime checks whether a JSP is current by comparing the modification times of the class and source files

EAServer 5.x `com.sybase.jaguar.webapplication.` property	EAServer 6.0 equivalent	Description
`keepgenerated`	Not applicable.	Enables preservation of Java source files that EAServer generates to create servlets when compiling the JSPs installed in this Web application
`large-icon`	<large-icon> in *web.xml*.	Specifies the name of the large icon file associated with the Web application.
`lazydistributedhttpsessionvalidation`	Not applicable.	Enables and disables lazy verification for distributed HTTP sessions.
`listeners`	<listener> in *web.xml*.	Specifies application life cycle event listeners installed in this Web application
`login-config`	<login-config> in *web.xml*.	Configures login authentication for the Web application.
`mime-mapping`	<mime-mapping> in *web.xml*.	Configures MIME mappings for the Web application to augment or override the server's default MIME mappings.
`name`	Not needed.	Web application name
`refresh`	Always refreshable.	Specifies whether the Web application can be refreshed.
`resource-env-ref`	<resource-env-ref> in *web.xml* and *bind:java:comp/env/jms/*<topic or queue>.	Resource environment references are logical names applied to objects administered by EAServer.
`resource-ref`	<resource-ref> in *application-client.xml* and *bind:java:comp/env/*<jdbc, jms, mail, url>	Specifies aliased JNDI names for database connections, JavaMail sessions, and URL factories used by the Web application.
`runasidentity.<id>`	<run-as> in *web.xml*.	Maps an identity name used in servlet properties to an identity defined in the EAServer repository.
`SAXfactory`	Not applicable. (Use JAXP1.2 standard procedures to specify).	SAX Parser factory class name.
`sectrace`	Not applicable.	Enables or disables security implementation tracing.
`security-constraint`	<security-constraint> in *web.xml*.	Associates required user roles and transport security for Web resource collections defined in the `com.sybase.jaguar.webapplication.web-resource-collection` property.

EAServer 5.x `com.sybase.jaguar.webapplication.` property	EAServer 6.0 equivalent	Description
`security-role.<j2ee-role>`	Role mappings in web application's user *config.xml*.	Specifies a mapping from a J2EE role name used in the Web application to a role defined in the EAServer repository.
`servlet-mapping`	<servlet-mapping> in *web.xml*.	Associates installed servlets with request paths.
`session-config`	<session-config> in *web.xml*.	Configures HTTP session properties.
`session-id`	Not applicable.	Specifies how a Web application sends and receives the HTTP session identifier. The identifier can be sent as a cookie, or it can be encoded in the URL. To encode the session identifier as part of the URL (also known as URL rewriting), servlet and JSP developers must call the HttpServletResponse.encodeURL(String) method or equivalent methods.
`sharecompiledjspclasses`	Not applicable.	Enables sharing of JSP class files by servers that run from the same EAServer installation
`small-icon`	<small-icon> in *web.xml*.	Specifies the name of the small icon file associated with the Web application
`taglib`	<taglib> in *web.xml*.	Configures path aliases for JSP Tag Library Descriptors (TLDs) used in the Web application.
`web-resource-collection`	<web-resource-collection> in *web.xml*.	Specifies a collection of request paths to be protected by security constraints
`welcome-file-list`	<welcome-file-list> in *web.xml*.	Configures welcome files for the Web application. Welcome files are used to satisfy HTTP requests that end in a directory name, rather than specifying the full path to a file or a path that is mapped to a servlet invocation.
`XSLTfactory`	Not applicable. (Use JAXP1.2 standard procedures to specify).	XSLT Parser factory class name.

Index

L

M

N

P

R

S

T

U

W