

# Getting Started with Oracle® Java CAPS

Copyright © 2009, 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

# Contents

---

<b>Getting Started with Java CAPS</b> .....	5
Java CAPS Runtime Execution .....	5
One-Time Setup Steps for Java CAPS .....	6
Quick Start Installation Instructions .....	7
Quick Start Installation Steps: Complete .....	7
Quick Start Installation Steps: Custom .....	8
Java CAPS Design Process Overview .....	9
End-to-End Step-by-Step Scenario .....	11
Overview .....	11
Before You Begin .....	13



# Getting Started with Java CAPS

---

This document provides a very high-level view of installing and working with Java CAPS. It includes the following topics:

- “Java CAPS Runtime Execution” on page 5
- “One-Time Setup Steps for Java CAPS” on page 6
- “Quick Start Installation Instructions” on page 7
- “Java CAPS Design Process Overview” on page 9
- “End-to-End Step-by-Step Scenario” on page 11

You can find samples and tutorials to help you work with Java CAPS at <http://java.net/projects/javacaps-samples/pages/Home>.

## Java CAPS Runtime Execution

The job of Java CAPS applications is typically to receive data from an external system, usually to manipulate or transform that data according to predefined rules, and then to send the data either back to its source or to a different external system. An external system could be a database, a file, a messaging queue, or application, such as Oracle Financials or SAP.

Java CAPS applications within the Java EE do the following:

- Get data from an external system. This usually requires one or more adapters, such as the File Adapter.
- Persist the data, using JMS topics and queues (usually).
- Transform the data, using one or more collaborations or business processes (usually).
- Communicate and orchestrate intermediate results amongst processes.
- Send the results to an external system or file. This usually requires one or more adapters.

Within the Java EE container, Java CAPS applications can use web services, but this is not required. However, to communicate with the overall JBI container, web services are required.

The services provided by the Java CAPS application are provided through the Java EE service engine; JBI uses WSDL to access them as web services.

JBI service units and service assemblies do the following:

- Get data from an external system, using one or more binding components such as the HTTP BC.
- Persist and communicate intermediate results, using the Normalized Message Router.
- Transform the data, using one or more service engines such as the BPEL SE or Java EE SE.
- Put results somewhere, using one or more binding components.

From the point of view of the CAPS application, the bridge to the Java EE Service Engine looks and acts like an adapter. From the point of view of the JBI container, CAPS applications are described by WSDL files and served by the Java EE service engine.

## One-Time Setup Steps for Java CAPS

1. Ensure your system meets the prerequisites. For complete information, refer to the installation documentation and Release Notes. Examples might include:
  - At least 2.5 GB free disk space for the installed files
  - At least 160 MB RAM for each repository server
  - At least 640 MB RAM for each machine running the NetBeans IDE
  - At least 640 MB RAM for each application server
  - JDK 1.6 update 20 (refer to the installation documentation for specific platform requirements)
2. Ensure you have access to all external systems with which Java CAPS will need to communicate. Examples might include:
  - Path names to local files
  - Passwords for secure FTP server
  - Hostname of Oracle database server
  - Oracle Advance Queues or WebLogic JMS properties
  - Parameters for communicating with SAP repository
3. Install Java CAPS, referring as needed to the installation documentation or the quick installation steps. The standard installation installs all JBI components, but not all Java EE components. You may need to perform additional post-installation steps to make sure all required products are installed.
4. As needed, configure external systems to communicate with Java CAPS. Specific steps vary widely from system to system. Examples might include:
  - Upload LDAP schema files
  - Run SQL scripts

- Create and configure PeopleSoft nodes
- Configure CICS security settings and connection logic
- Install and initialize remote APIs for SWIFT AG

## Quick Start Installation Instructions

You can install Java CAPS in either of two ways: In the complete style, where all base products are installed, or in a custom style, where you choose which base components to include.

- [“Quick Start Installation Steps: Complete” on page 7](#)
- [“Quick Start Installation Steps: Custom” on page 8](#)

### Quick Start Installation Steps: Complete

This summary assumes you will perform a complete installation of all base products on a single Windows machine. It also assumes that you might optionally also install additional products on the same machine. Within the base products, only the Repository, NetBeans IDE, and Application Server are required for design-time operations. However, it is recommended that you install all base products. For example, installing Enterprise Manager at this point in the process allows you to access tools for monitoring and administration.

- To install the base products, you run Java CAPS executable file for your platform and, in the installation wizard, supply values for various parameters.
- To install additional Repository-based products, you start the Repository, launch the Java CAPS Suite Installer, and upload one or more SAR files.

#### ▼ To Install Java CAPS Base Products

Performing a complete installation of base components on a single Windows machine (minimum 2.0 GB RAM) requires approximately 30 minutes.

- 1 **On your installation media, double-click the Java CAPS executable file for your operating system to start the installation wizard.**
- 2 **Click Next and Back to step through the wizard, and supply site-specific parameter values or retain the default parameter values.**

Parameters include port numbers, passwords, installation location, and so on.

#### ▼ To Install Additional Repository-Based CAPS Products and Components

The following steps are intended to summarize. For details on specific products and components, refer to the installation documentation.

- 1 **Open the Java CAPS home directory.**
- 2 **Start the repository by double-clicking the `start_repository` shortcut.**
- 3 **Start a new browser session and point your browser at the URL for the Java CAPS Uploader: `http://localhost:12000`.**
- 4 **Log in to the Uploader:**
  - Username: `admin`
  - Password: `adminadmin`
- 5 **Use the Administration tab to install additional products in the form of SAR files, as needed. These files are located in the components folder of your installation media.**
  - The `Adapters` folder includes adapters for CICS, COM, LDAP, TCPIP, and many others.
  - The `Addons` folder includes: Alert Agent and SNMP Agent, COBOL Copybook Converter, eIndex and eView, and migration tools, plus the MDM solution templates.
  - The `ESB API Kit` folder includes API Kits for many operating system platforms.
  - The `Message Libraries` folder includes message libraries for EDIFACT, HIPAA, HL7, X12, and others.
- 6 **As needed, use the Downloads tab to load files locally so you can extract their contents.**

This concludes the steps for installing Java CAPS. To exercise them, you can try out any of the repository-based end-to-end scenarios supplied with Java CAPS or the various adapters.

You can download sample projects to work with from the Java CAPS sample site at <http://java.net/projects/javacaps-samples/pages/Home>.

## Quick Start Installation Steps: Custom

This summary assumes you will perform a custom installation of the NetBeans IDE and Application Server on a single Windows machine. Performing a custom installation allows you to choose which of the Java CAPS components to install. Performing a custom installation of these base components on a single Windows machine (minimum 1.5 GB RAM) requires approximately 15 minutes.

### ▼ To Custom-Install Java CAPS

- 1 **On your installation media, double-click the Java CAPS executable for your operating system to start the installation wizard.**
- 2 **At the Choose Install Set step, select the Custom option and then click Next.**



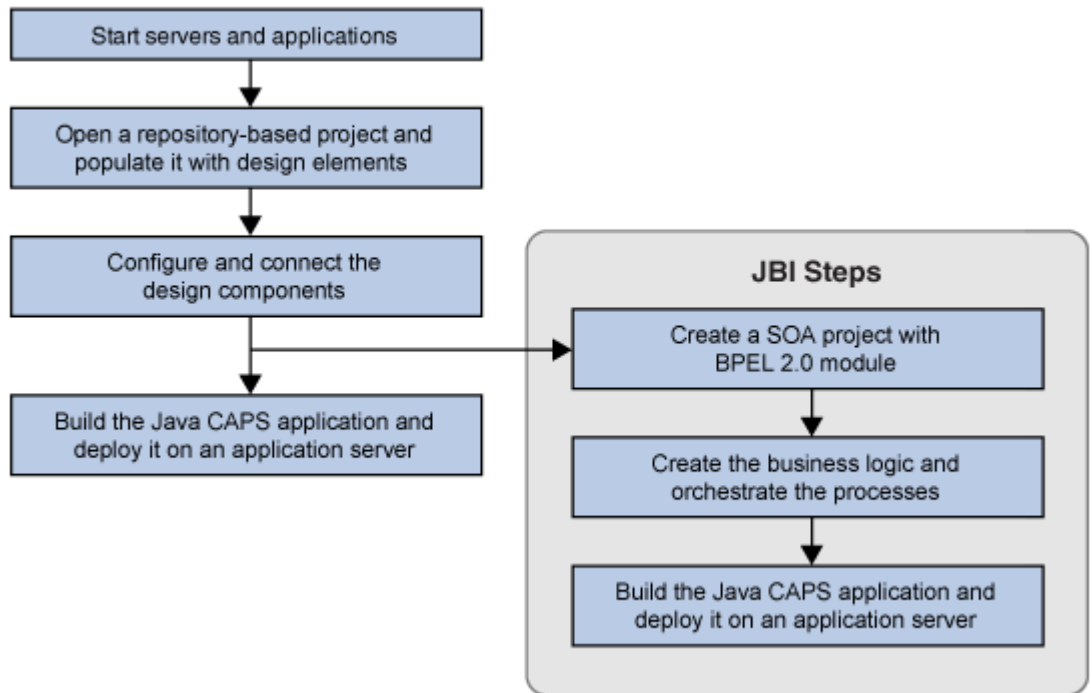
- 3 **Supply site-specific parameter values, or retain default parameter values.**

## Java CAPS Design Process Overview

Designing a repository-based Java CAPS project requires the following tasks:

- Start the necessary internal and external servers.
- Start the NetBeans IDE and open a new or existing CAPS project.
- Create, configure, and connect design elements.
- As needed, create business logic and orchestrate business processes, Collaborations, or page flows.
- Create a CAPS environment and deployment profile.
- Build the CAPS application, and then deploy and test it on an application server.
- As needed, bridge the CAPS application to JBI and combine it with other projects to create a composite application. Build the composite application and deploy/test it on an application server.

## Overview of Design Process: Repository Based



Java CAPS repository-based projects can include the following design elements, each with its own type of editor:

- Properties to be used for design-time components such as adapters, engines, and queues
- Properties to be used for runtime servers for external and internal systems
- Message libraries and object type definitions (OTDs) to model and encode data
- Java collaboration definitions (JCDs) to transform data
- Other data-transformation engines, such as for eTL collaborations or XSLT transformations
- Business processes (BPs) and page flows to implement business-logic rules and to orchestrate the execution of components and web services. Within Java CAPS projects, the editor uses Business Process Execution Language (BPEL) 1.0.
- Other specialized design elements, such as page layouts or cascading style sheets (CSSs)
- Connectivity maps (CMaps) to establish and configure connections between design-time components
- Deployment profiles (DPs) to assign engines, adapters, and queues with corresponding server resources

Composite applications can include the following additional design elements, each with its own type of editor:

- Properties to be used for Binding Components (BCs) and Service Engines (SEs)
- Business logic and orchestration. In the JBI container, composite applications use BPEL 2.0.
- Web Service Description Language (WSDL) elements with editable message definitions, port types, partner links, and so forth
- Composite Application Service Assembly (CASA) configurations between JBI module connections with WSDL ports and external modules

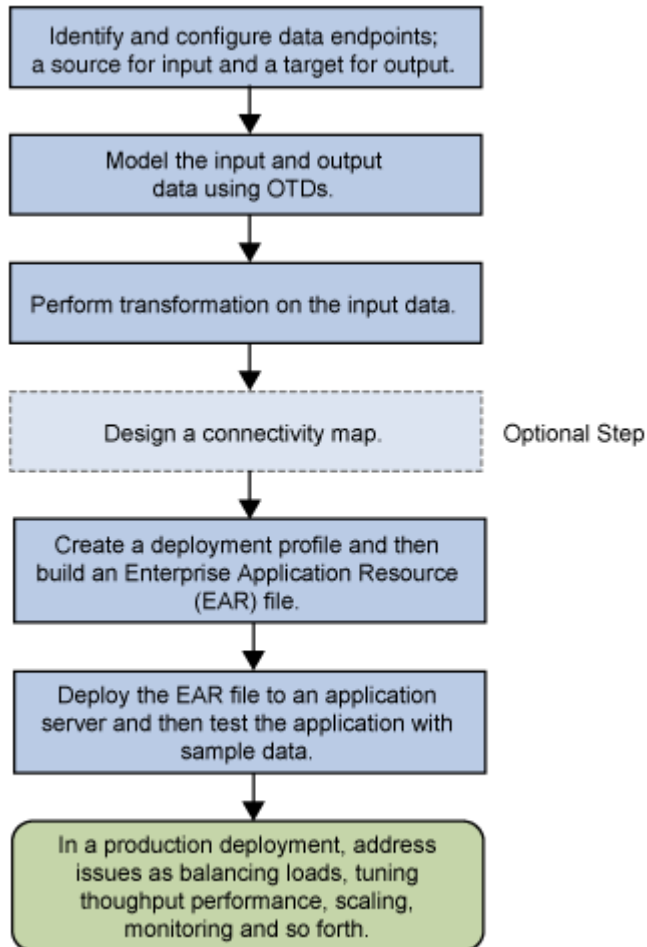
## End-to-End Step-by-Step Scenario

This section provides an overview of the steps required to create a simple application without requiring any special external systems. The simplicity of the application's runtime execution is disproportionate to the tools you use to design it. This is because the purpose of the scenario is instructive rather than functional: It shows you how to exercise the main design components of Java CAPS and how to use a Java CAPS application in a JBI container.

### Overview

1. Identify and configure data endpoints: a source for input and a target for output.
  - In a simple end-to-end scenario, you might use a file for input data records and JMS for output data records.
  - In a real-world application, you might use several input and output streams encompassing multiple external systems. You would not use JMS as a primary endpoint, but rather as a conduit for persisting data.
2. Model the input and output data using OTDs.
  - In a simple end-to-end scenario, you might create a user-defined OTD for input and an XSD-based OTD for output.
  - In a real-world application, you might use OTDs from a specialized prebuilt message library, or you might use OTDs corresponding to database adapters and/or communication adapters.
3. Perform a transformation on the input data.
  - In a simple end-to-end scenario, you might copy data using getter and setter methods, perform a comparison, and insert a value based on a variable.
  - In a real-world application, you might well do just a simple copy to translate between disparate systems, or you might instead create many modules that perform hundreds or thousands of operations on the data.
4. Design a simple business process.

- In a simple end-to-end scenario, you might use a very simple Receive-Activity-Reply pattern.
  - In a real-world application, you might not use business processes at all, or you might use multiple complex business processes with complex logic.
5. Design a connectivity map.
    - In a simple end-to-end scenario, you might create and populate a connectivity map, connect the components, and configure the connections.
    - In a real-world application, you might use multiple connectivity maps.
  6. Create a deployment profile and then build an Enterprise Application Resource (EAR) file.
    - In a simple end-to-end scenario, you might assign the components in one connectivity map to a few internal systems and one external system (File).
    - In a real-world application, you might use multiple connectivity maps, and you would usually have several different external systems.
  7. Deploy the EAR file to an application server and then test the application with sample data.
    - In a simple end-to-end scenario, you might feed sample data one record at a time and check output.
    - To test the design of a real-world application, you might need to feed several different kinds of sample data from several systems, you would try out a wide variety of positive and negative test cases, and iteratively refine your design.
  8. (In a production deployment, you would also address such issues as balancing loads, tuning throughput performance, scaling, monitoring, and so forth.)



## Before You Begin

Before you start working with projects, you need to perform the following steps.

1. Navigate to your Java CAPS home directory.
2. Start the two servers you need by running these two shortcuts:
  - `start_repository`
  - `start_appserver_domain1`
3. Start the NetBeans IDE by running this shortcut:
  - `start_netbeans`

You are not ready to create projects using the NetBeans IDE. You can practice creating your own projects, or you can try one of the sample projects provided for Java CAPS. You can download sample projects for various Java CAPS components from the Java CAPS sample site at <http://java.net/projects/javacaps-samples/pages/Home>.

If you are an experienced Java CAPS 5.x designer, take time to acquaint yourself with the differences between the new NetBeans IDE and the Enterprise Designer GUI from previous releases:

- The GUI has a different look and feel from version 5.x. Experiment with panes that can be independently docked, hidden, and rearranged, and acquaint yourself with such new features as macros, source editors, tag libraries, keyboard shortcuts, refactoring options, versioning tools, and so forth. Online help applies to the entire NetBeans IDE, not to CAPS-specific design components.
- Java CAPS projects are now just one of many different project types, and projects that are not currently in use are not displayed.
- Java CAPS environments are now subsumed into a larger category of Services.