

# Oracle9iAS Plug-in for Borland JBuilder

Readme for Installation, Configuration, and Use

Release 2 (9.0.3) for Borland JBuilder 8

November 2002

**Part No. B10444-01**

## Introduction

Oracle9iAS Plug-in for Borland JBuilder allows Borland JBuilder developers to create and distribute their applications with Oracle9iAS Containers for J2EE (OC4J). OC4J is a complete Java2 Enterprise Edition (J2EE) container that supports several Java specifications, such as servlets, JavaServer Pages (JSP), and Enterprise JavaBeans (EJB). Using this plug-in, developers can create applications with Borland JBuilder and then distribute those applications with one or more OC4J servers.

## Prerequisite Software

You must have installed the following products on your system before using this plug-in:

- **Borland JBuilder 8 Enterprise Edition**

For product information and documentation about Borland JBuilder, visit the Borland JBuilder Web site at

<http://www.borland.com/jbuilder/>

- **Oracle9iAS Containers for J2EE release 2 (9.0.3)**

For information about OC4J, visit the Oracle Technology Network Web site at

<http://otn.oracle.com/index.html>

**ORACLE®**

Copyright © 2002, Oracle Corporation.  
All Rights Reserved.

Oracle is a registered trademark, and Oracle9i is a trademark or registered trademark of Oracle Corporation. Other names may be trademarks of their respective owners.

For more information about OC4J, click **Oracle9i Application Server** from the **Products** link in the left-hand panel. Then, click the **Oracle9iAS Containers for J2EE** link in the right-hand panel.

This plug-in is supported on the following operating systems:

- Solaris Operating System
- Windows
- Linux

This document assumes basic familiarity with Borland JBuilder and OC4J.

---

---

**Note:** By default, Borland JBuilder and OC4J use port 8888. To avoid conflicts, select a different port for your OC4J Web site.

---

---

## Installing the Plug-in

To install the plug-in, follow these steps:

1. Extract the contents of the `OC4J903JB8P.zip` file to a temporary directory.
2. Locate the `OC4J903JB8.jar` file from the extracted ZIP file. Then, place it in the `lib/ext` subdirectory under the Borland JBuilder home directory.
3. Stop and restart Borland JBuilder.

## Configuring OC4J

After installing the plug-in, you must configure the OC4J server, the deployment targets, and the OC4J libraries. The following sections describe each of these tasks.

### Configuring the OC4J Server

To configure the OC4J server, follow these steps:

1. Select **Configure Servers...** from the **Tools** menu.  
The **Configure Servers** screen appears.
2. Select **Oracle9i Application Server 9.0.3** from the **User Home** directory.
3. Click the **General** tab on the **Server Settings** panel. Then, select the **Home directory** field. Use the browse button to identify the home directory for OC4J (usually `$ORACLE_HOME/j2ee/home`).

4. Click the **Custom** tab on the **Server Settings** panel. Then, select the **Admin password** field. Enter the administration password you defined when you installed OC4J.

## Configuring the Deployment Targets

To configure the deployment targets, follow these steps:

1. Deployment targets can be local or remote. You can specify a default target for J2EE applications, configure new OC4J deployment targets, modify existing targets, or delete existing targets as follows:
  - The **Default deployment target** field shows the current default deployment target. Select a different deployment target from the **OC4J Deployment Targets** list and click **Default**.
  - To configure a new deployment target or modify an existing target, click **New** or **Modify**, respectively.

The **Oracle9iAS Containers for J2EE - Properties** dialog box appears.
  - To delete an existing target, click **Delete**.
2. Enter the appropriate configuration information in the following fields:

---

---

**Note:** By default, some of these fields contain initial settings. You can accept the current setting for each field or change it, as needed.

---

---

- **Container name** - A unique user-defined name for the deployment target (for example: OC4JDEV or OC4JPROD1, OC4JPROD2, OC4JPROD3, and so on). This field is required.
- **Server version** - The version number for the OC4J server (for example: 9.0.3). Enter the following command to obtain the version number from the OC4J server installation:

```
java -jar oc4j.jar -version
```

This field is required.

- **Server host name** - The host name for the OC4J server installation (for example: localhost or oc4jhost.us.oracle.com). This field is required.
- **Server RMI port** - The RMI port for the OC4J server. This information is available from the OC4J installation. Specifically, this attribute is defined in `rmi.xml` in the `<oc4j_home>/config/` directory. (The `rmi.xml`, or remote method invocation file,

contains configuration information for calling remote methods.)  
The default value for this port is 23791. This field is required.

- **User name** - The user name you specify for application deployment. This field is required.
  - **Password** - The password that accompanies the user name. This field is required.
  - **Target path (opt)** - The path on the server node on which to deploy your applications. The default is `Null`. This field is optional.
3. When you have finished entering the OC4J container properties, click **OK** to save the new settings. Your new settings are shown on the **Oracle9i Application Server** tab page.

## Configuring the OC4J Libraries

Before you can test or debug EJB client applications from Borland JBuilder, you must make the OC4J libraries available for Borland JBuilder for each project.

To create a new OC4J library, follow these steps:

1. Select **Configure Libraries** from the **Tools** menu.
2. From the **Configure Libraries** screen, click **New** to display the **New Library Wizard**.
3. Enter `OC4J` in the **Name** field.
4. Identify the `oc4j.jar` file in the home directory for OC4J and click **OK** to save the setting.

The **Configure Libraries** screen shows the name you entered.

## Setting Up Project Properties

You can select Oracle9iAS for an existing project or as the default for all future projects. The following sections describe each process.

### Specifying Properties for Existing Projects

To select Oracle9iAS for an existing project, follow these steps:

1. Select **Project Properties...** from the **Project** menu.  
The **Project Properties** screen appears.
2. Select the **Server** tab.
3. Select **Single server for all services in project**.

4. Select the appropriate server from the pull-down list (for example: Oracle9i Application Server 903).
5. Click **OK**.

## Specifying Properties for Future Projects

To select Oracle9iAS as the default for future projects, follow these steps:

1. Select **Default Project Properties...** from the **Project** menu.  
The **Default Project Properties** screen appears.
2. Follow the procedure described previously in [Specifying Properties for Existing Projects](#) to make your changes and save them.

## Setting Up Deployment Properties

Before you can deploy an application to OC4J, you must create an enterprise archive (EAR) for the application, using the JBuilder EAR Wizard. Optionally, you can specify the OC4J deployment properties for the application.

## Specifying the OC4J Deployment Properties

To set up the deployment properties for your application, follow these steps:

1. Right-click an EAR file or its corresponding group (.eargrp) and select **Properties**.  
The **Properties** screen appears.
2. Select the **Deployment** tab.  
The **OC4J Deployment Properties** screen appears.
3. Review or modify the following fields as needed:
  - **Container name** - The name of the container you configured as the deployment target (for example: OC4JDEV). Select the desired deployment target from the pull-down menu. This field is required.
  - **Host name** - The host name for the OC4J server, which is obtained from the container configuration. This field cannot be modified.
  - **User name** - The user name specified for application deployment. This field can be modified.
  - **Password** - The password that accompanies the user name. This field can be modified.

- **Target path (opt)** - The path on the server node on which to deploy your applications. This field is optional.
  - **Deployment name** - The user-defined application name. By default, the project name for the application is inserted in this field. (The project name is the name used to save the application in Borland JBuilder.) This field can be modified.
  - **Web site name** - The Web site to which your Web applications are bound. This field is used only if the EAR file you want to deploy contains Web applications.
4. Click **OK** to save the property settings.

## Creating or Modifying the Vendor-Specific Deployment Descriptor File

When you use Oracle9iAS for your Enterprise JavaBeans project, the Borland JBuilder plug-in lets you create and maintain a vendor-specific deployment descriptor file (`orion-ejb-jar.xml`), along with the J2EE standard deployment descriptor file (`ejb-jar.xml`). You can edit these files manually or use the graphical user interface (GUI) provided with this plug-in.

To display the contents of the deployment descriptor files, click the **EJB DD Source** tab in the **Deployment Descriptor Editor** screen.

The selected deployment descriptor file is displayed in the window panel on the right side of the screen. From there, you can edit the file manually or use the GUI to modify it.

To access the GUI, follow these steps:

1. Select a node from the left-hand panel and double-click it.  
The deployment descriptor file is displayed in the right-hand panel.
2. Open the directory for the selected node to select an entity bean, session bean, or Message Driven Bean (MDB). Then, double-click the selected bean.  
The right-hand panel displays the selected component and the **General** tab.
3. Click the **Oracle9iAS 903 Properties** tab at the bottom of the right-hand panel.

The right-hand panel displays the properties for the selected component. From there you can modify the values of these properties in the **Value** column by selecting each attribute and entering new values or editing existing values.

## Creating the OC4J Web Service

The Borland JBuilder plug-in lets you create and specify attributes for the OC4J Web service. To create the service, follow these steps:

1. Select a node. Choose either the specified node for your .java file or a stateless session bean node.
2. Select **Export Class as an Oracle9iAS Web Service** from the mouse right-click menu.

The **OC4J Web Service** screen appears.

3. Enter the appropriate information in the following fields:
  - **Service Name** - A user-defined name for the Web service (for example: `WebService1`). This field is required.
  - **War** - The Web Application Archive (WAR) file to be used to create the Web service. You can select the WAR file name from the pull-down list. This field is required.
  - **Stateful** - The state of the Web service session. You can select a stateful session or accept the default, stateless. This field is required.
  - **Interface** - The Java interface that you can specify for your Web service. This field is optional.
4. Click **OK**.

See *Oracle9i Application Server Web Services Developer's Guide* for more information about developing and deploying Web services.

## Deploying and Undeploying Applications on OC4J

You can select one of two deployment options for your application on OC4J. If you select Deployment Option 2, you can also undeploy your application. The following sections describe each option.

### Deployment Option 1

This option lets you use the OC4J Deployment Tool to deploy the application to one or more targets. Follow these steps:

1. Verify that the target server is running, or start the server.
2. Select **Enterprise Deployment** from the **Tools** menu.

The **OC4J Deployment Tool** screen appears.

3. Modify the deployment properties as described previously in [Specifying the OC4J Deployment Properties](#).
4. In addition to the deployment properties, you can select an EAR or WAR file to deploy. To deploy the application, select the desired EAR or WAR file from the pull-down menu in the **EAR file to deploy** field, then click **OK**.

The OC4J Deployment Tool displays a confirmation message or an error message from the OC4J server.

## Deployment Option 2

This option lets you use the mouse right-click submenu within Borland JBuilder to deploy the application to one or more targets or undeploy the application. Follow these steps:

1. Verify that the target server is running, or start the server.
2. Right-click an EAR or WAR file or its corresponding group (.eargrp or .wargrp) and select **Deploy**, **Redeploy**, or **Undeploy** from the **Deploy Options** submenu.

The OC4J server displays status and confirmation messages in a view panel at the bottom of the Borland JBuilder window.

## Running and Debugging EJB Applications

To run and debug EJB applications deployed on a local OC4J server, follow the instructions in the Borland JBuilder documentation available at the Borland JBuilder Web site at

<http://www.borland.com/jbuilder/>

---

---

**Note:** Before you can run or debug an OC4J client from Borland JBuilder, you must include the OC4J library as one of the required libraries.

---

---

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology



vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>

**Accessibility of Code Examples in Documentation** JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

**Accessibility of Links to External Web Sites in Documentation** This documentation may contain links to Web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

