

# **Deploying a COOL:Joe 1.1 (Service Pack 1) Enterprise JavaBean to the JRun 3.0 Server – Enterprise Edition**

**Author:** Sreedhara Narayanaswamy

**Version:** 1.0

**Date:** October 20<sup>th</sup>, 2000

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. at any time.

THIS DOCUMENTATION MAY NOT BE COPIED, TRANSFERRED, REPRODUCED, DISCLOSED OR DUPLICATED, IN WHOLE OR IN PART, WITHOUT THE PRIOR WRITTEN CONSENT OF COMPUTER ASSOCIATES INTERNATIONAL, INC. THIS DOCUMENTATION IS PROPRIETARY INFORMATION OF COMPUTER ASSOCIATES INTERNATIONAL, INC. AND IS PROTECTED BY THE COPYRIGHT LAWS OF THE UNITED STATES AND INTERNATIONAL TREATIES.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, COMPUTER ASSOCIATES INTERNATIONAL, INC. PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL COMPUTER ASSOCIATES INTERNATIONAL, INC. BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL OR LOST DATA, EVEN IF COMPUTER ASSOCIATES INTERNATIONAL, INC. IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

THE USE OF ANY PRODUCT REFERENCED IN THIS DOCUMENTATION AND THIS DOCUMENTATION IS GOVERNED BY THE END USER'S APPLICABLE LICENSE AGREEMENT.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with *Restricted Rights* as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227.7013(c)(1)(ii) or applicable successor provisions.

© 2000 Computer Associates International, Inc., One Computer Associates Plaza, Islandia, New York 11749. All rights reserved.

All product names referenced herein belong to their respective companies.

Some of the other designations used in this document are trademarks of other companies not associated with Computer Associates International, Inc. Such use of these trademarks is for information purposes only and is not intended to be generic in nature or to imply that such trademarks are owned by or licensed to Computer Associates International, Inc.

## Table of Contents

Preface.....	4
Prerequisites .....	4
Definitions.....	4
Configuring JRun3.0.....	5
Creating a Datasource in the JRun Management Console.....	5
Data Source Examples .....	6
Classpath Examples .....	9
Editing the deploy.properties File.....	9
Building the COOL:Joe 1.1 SP1 Test Harness .....	13
Starting the JRun EJB Engine .....	14

## Preface

This paper was written to provide detailed information about deploying an EJB generated with COOL:Joe version 1.1 Service Pack 1 (SP1) to JRun version 3.0 Application Server Enterprise Edition. This document targets the Windows NT platform for the English language version only. The information in this document is subject to change as updates and enhancements are made to the software. For clarifications on specific topics, please contact the Customer Support team.

## Prerequisites

Using the procedures described in this paper assumes that you have completed certain tasks using COOL:Joe 1.1 with Service Pack 1.

You must have implemented your component up to and including the point at which you generate the test harness and/or Web Application. More information on Workflows for using COOL:Joe 1.1 is available in the online help, or through *Task Advisor* and the task definitions provided with COOL:Joe.

Specifically, you must have:

- Install the JRUN3.0 plugin for COOL:Joe1.1 SP1 and perform the installation as specified in the readme file
- Created a component implementation
- Defined business logic
- Generated and built an EJB Session Bean
- Created a default project for the EJB
- Generated a Test Harness and/or Web Application
- Created a default project for the Test Harness and/or Web Application

Do *not* build the Test Harness or Web Application with the *Build Wizard* prior to running the procedures in this document!

## Definitions

EJB - Enterprise JavaBeans

JMC - JRun Management Console

JDBC - Java Database Connectivity

JNDI - Java Naming and Directory Interface

## Configuring JRun3.0

### *Creating a Datasource in the JRun Management Console*

1. Start the JRun Management Console from the Windows **Start** menu and log in. The default userID and password are both “admin”.
2. a) In the left panel of the JMC window, click the *JRun Admin Server* entry and then expand the tree to display choices for that server  
b) The main or central panel of the window will show server statistics. If the *Server Status* field shows a value of “stopped”, you will need to click the **restart server** button.
3. Click the *JDBC Data Sources* entry.

This displays the *JDBC Data Source* panel in the main or central panel of the window. In this area, you can add, edit, and delete data source entries.

4. Click the **edit** button in the main panel to open the *Edit Window* and add your data source using the definitions below as a guide.

When adding or editing a data source, these properties are available:

Name	The name of the data source. This is the name that you will use to reference the data source in your servlet.
Display Name	The data source name as it will be displayed in the JRun Management Console.
Driver	The name of the JDBC driver to load. You need to ensure that the JDBC driver is available on the CLASSPATH. You can modify the current CLASSPATH through the JRun Management Console using the Java settings for the server.
URL	The JDBC URL used when connecting to the database. See the documentation for the JDBC driver for details and an explanation of options.
Description	The description of the data source that will be displayed in the JRun Management Console.
Pooling	If this check box is marked, pooling will be enabled for this data source.

Timeout	The number of <i>seconds</i> of inactivity before JRun will close the connection. The default is 300 seconds. (In the current version of both the JRun Management Console and the user documentation, this value is incorrectly stated as being in minutes.)
Interval	How often (in seconds) a background thread will clean up any stale connections. The default is 60 seconds.
Username	The user name for the database (if required).
Password	The password for the database (if required).
Vendor Arguments	Name/value pairs for vendor-specific arguments. Some database products allow custom parameters and options to be set using JDBC properties. See the vendor documentation for your JDBC driver for information about custom arguments.

### Data Source Examples

#### For MSSQL

<b>Name</b>	<JNDI Name>
<b>Display Name</b>	<Display Name>
<b>Driver</b>	com.inet.tds.TdsDriver
<b>URL</b>	jdbc:inetdae:<MSSQL-SERVERNAME>:<MSSQL-Server portno>?database=<Database-Name>
<b>Pooling</b>	Enable it by clicking the radio-button
<b>Username</b>	<user-id>
<b>Password</b>	<password>

<b>Name</b>	<JComp2>
<b>Display Name</b>	<JComp2>
<b>Driver</b>	com.inet.tds.TdsDriver
<b>URL</b>	jdbc:inetdae:snsrver:1433?database=SNJRUN
<b>Pooling</b>	Enable it by clicking the radio-button
<b>Username</b>	<sa>
<b>Password</b>	

For Oracle

<b>Name</b>	<JNDI Name>
<b>Display Name</b>	<Display Name>
<b>Driver</b>	oracle.jdbc.driver.OracleDriver
<b>URL</b>	jdbc:oracle:thin:@<Oracle Server name>:<Oracle Server Port no>:<Database name>
<b>Pooling</b>	Enable it by clicking the radio-button
<b>Username</b>	<user-id>
<b>Password</b>	<password>

<b>Name</b>	<JComp2>
<b>Display Name</b>	<JComp2>
<b>Driver</b>	oracle.jdbc.driver.OracleDriver
<b>URL</b>	jdbc:oracle:thin:@snoracle:1521:ORCL
<b>Pooling</b>	Enable it by clicking the radio-button
<b>Username</b>	<system>
<b>Password</b>	<manager>

For DB2/2

<b>Name</b>	<JNDI Name>
<b>Display Name</b>	<Display Name>
<b>Driver</b>	COM.ibm.db2.jdbc.app.DB2Driver
<b>URL</b>	jdbc:db2:<Database name>
<b>Pooling</b>	Enable it by clicking the radio-button
<b>Username</b>	<user-id>
<b>Password</b>	<password>

<b>Name</b>	<JComp2>
<b>Display Name</b>	<JComp2>
<b>Driver</b>	COM.ibm.db2.jdbc.app.DB2Driver
<b>URL</b>	jdbc:db2:snsdb
<b>Pooling</b>	Enable it by clicking the radio-button
<b>Username</b>	Dbuser1
<b>Password</b>	Dbuser1

For Cloudscape

<b>Name</b>	<JNDI Name>
<b>Display Name</b>	<Display Name>
<b>Driver</b>	COM.cloudscape.core.JDBCdriver
<b>URL</b>	jdbc:cloudscape:<Database Name>
<b>Pooling</b>	Enable it by clicking the radio-button
<b>Username</b>	<user-id> [leave it blank]
<b>Password</b>	<password> [Leave it blank]

<b>Name</b>	<JComp1>
<b>Display Name</b>	<JComp1>
<b>Driver</b>	COM.cloudscape.core.JDBCdriver
<b>URL</b>	jdbc:cloudscape:CloudscapeDB
<b>Pooling</b>	Enable it by clicking the radio-button
<b>Username</b>	<user-id> [leave it blank]
<b>Password</b>	<password> [Leave it blank]

5. Click the **update** button on the *Edit Window*. This creates the data source for that target database.
6. In the left panel, click *Java Settings* (for the JRun Admin Server) and then click Classpath in the main window. Enter the entire JDBC driver path:

## Classpath Examples

For **MSSQL**: <drive-name>:/inet

For **Oracle**: <drive-name>:/orant/jdbc/lib/classes12.zip

For **DB2/2**: <drive-name>:/SQLIB/java/db2java.zip

For **Cloudscape**:

<drive-name>:/j2sdkee1.2.1/lib/cloudscape/cloudscape.jar

7. Save the classpath by clicking the **update** button in the *Edit Window*.
8. Click the JRun Admin Server in the left panel of the JMC window and click the "restart server" button in the JRun Admin Server window & login again.
9. Select JDBC Datasources and click the **Test** button for the datasource you created. If the configuration is successful, JRun will return the message, The test for this data source was successful.
10. Repeat steps 2 b - 9 for the *JRun Default Server* entry in the left panel.

## Editing the *deploy.properties* File

1. Go to **Start** → **Settings...** → **Control Panel** → **System** and enter this line in the *Environment* tab's *User Variables* settings:

```
JRUN_HOME <drive-name>:\Allaire\JRun
```

2. Create a new *deploy.properties* file for JRun in <JRUN\_HOME>\servers\default\deploy directory and create the following entries:

```
ejipt.classServer.host=localhost
ejipt.classServer.port=2323

ejipt.ejbJars=<EJB.JAR FILE>
ejipt.userHomeName=default.UserHome
ejipt.roleHomeName=default.RoleHome
ejipt.loginSessionHomeName=default.LoginSessionHome
create.ejb.allowedIdentities=all
<Method-name>.ejb.allowedIdentities=all

ejipt.jdbcSources=<Data source name>
<Data source name>.ejipt.sourceDriverClassName=<JDBC Driver class>
<Data source name>.ejipt.sourceURL=<JDBC URL>
<Data source name>.ejipt.sourceUser=<user-id>
<Data source name>.ejipt.sourcePassword=<password>
ejipt.logStackTrace=true
```

An explicit example of the *deploy.properties* file with MSSQL JDBC entries follows:

```
##
# either delete the line below or replace 'localhost' with your
# host's name or address to allow remote clients to access the server

ejipt.classServer.host=localhost
ejipt.classServer.port=2323

ejipt.ejbJars=JComp2Pkg.jar
ejipt.userHomeName=default.UserHome
ejipt.roleHomeName=default.RoleHome
ejipt.loginSessionHomeName=default.LoginSessionHome
create.ejb.allowedIdentities=all
Checkmethod.ejb.allowedIdentities=all

ejipt.jdbcSources=JComp2
JComp2.ejpt.sourceDriverClassName=com.inet.tds.TdsDriver
JComp2.ejpt.sourceURL=jdbc:inetdae:NARSR01SRV:1433?database=SJRun
JComp2.ejpt.sourceUser=sa
JComp2.ejpt.sourcePassword=
ejipt.logStackTrace=true
```

<p><b>Note:</b> In this example, <b>Checkmethod</b> is the only method used in the EJB. You must specify <i>each</i> of the methods in your EJB with a line similar to the one shown to satisfy the security access for the JRun container. If you have more than one method in your EJB, you will have multiple lines similar to the <b>Checkmethod</b> entry.</p>
---

**JComp2Pkg.jar** is the EJB jar file which will be deployed, and **JComp2** is the datasource name which was created in the JMC Admin server and default server, **com.inet.tds.TdsDriver** is the MSSQL JDBC Driver, **jdbc:inetdae:NARSR01SRV:1433?database=SJRun** is the MSSQL JDBC URL where **NARSR01SRV** is the MSSQL Server name, **1433** is the port number, and **SJRun** is the MSSQL database. The **sa** is the userID.

## Configuring COOL:Joe 1.1 SP1

1. Start COOL:Joe 1.1 SP1 and open the model that contains your EJB.
2. Select **Tools** → **Preferences**, expand **Wizards**, and then select **Deployment**.
3. Set the Deployment preferences for COOL:Joe using these values. A sample panel is shown in Figure 1.

<b>Initial Context Factory</b>	allaire.ejpt.ContextFactory
<b>Provider URL</b>	ejpt://<JRUN Server name>:2323
<b>JDBC Driver</b>	leave this value blank
<b>JDBC URL</b>	java:comp/env/jdbc/<JNDI name>
<b>JDBC User ID</b>	leave this value blank
<b>JDBC Password</b>	leave this value blank

4. Click OK.

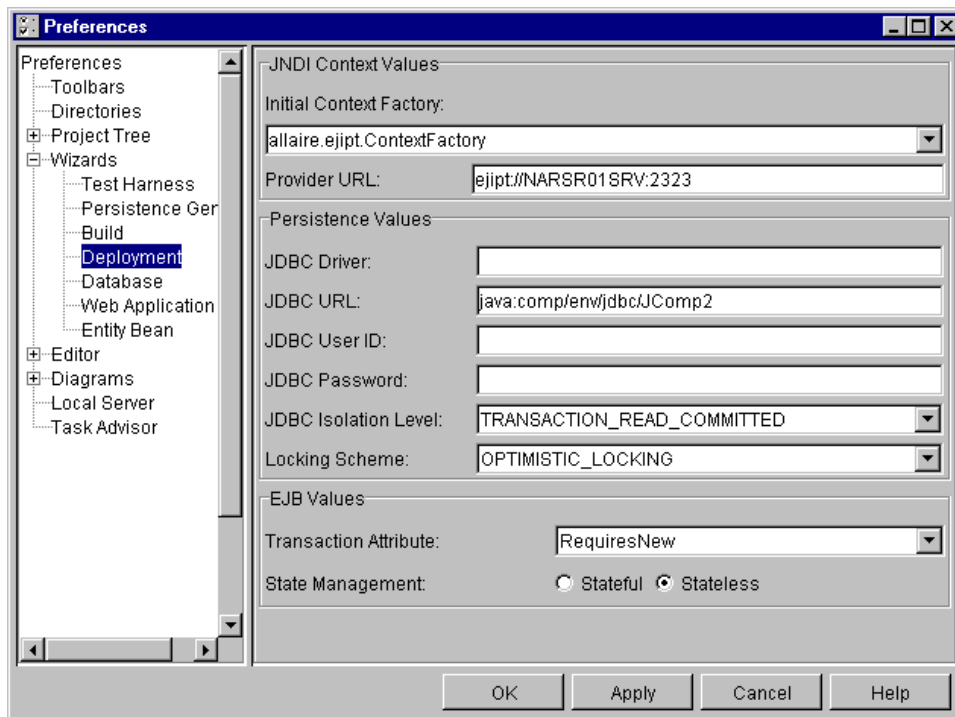






Figure 1 COOL:Joe 1.1 SP1 Deployment Preferences Wizard

**Note:** In this example, the JNDI name must be the same as the data source created in the Java Management Console and in the *deploy.properties* file.

## Deploying and Executing the EJB

1. In the COOL:Joe 1.1 SP1 Project Tree, select the  project which contains your EJB jar and build it.
2. Right-click on the  project and select  **Deploy EJBs...** to start the *Deployment Wizard*.
3. Click **Next** and select JRun3.0 Application Server from the *EJB Server Type* dropdown list.
4. Click  to create a profile for the JRun EJB Server, then click **Next**.

This shows the following default entries:

<b>Profile Name</b>	My JRUN Server1
<b>EJB Server Host Name</b>	Localhost
<b>EJB Server Port Number</b>	8001



*MY JRUN Server1* is the profile name. *Localhost* is the JRun server hostname and it could also be the JRun server hostname. *8001* is the JRun Server port number specified at the time of JRun installation.



5. Change the default EJB Server Port Number to 8001, or to the port setting used when JRun was installed.
6. Click **Finish** to deploy the EJB to the JRun server. You will get the message that the deployment was successful.

The deployment process creates the following JARs and files.

<b>File Name</b>	<b>Description</b>
ejipt_exports.jar	This file contains the stubs for the object and any additional classes that will be exported to clients on demand. This file is created during the deployment and is copied to the runtime directory.
ejipt_objects.jar	This file contains the remote and home interface implementation (EJB objects). This file is created during the deployment and is copied to the runtime directory.
runtime.properties	This file is generated during deployment in the deploy directory and is copied into the runtime directory.

## Building the COOL:Joe 1.1 SP1 Test Harness

1. From COOL:Joe 1.1 SP1 Project Tree, right-click on the test harness project, and select  **Properties....**
2. Click the  **Edit classpath** icon and add the following jars:  

```
<JRUN_HOME>\lib\ejipt_client.jar  
<drive-name>:\cool\joe\output\<EJBProject.jar>  
<JRUN_HOME>\servers\default\deploy\ejipt_exports.jar  
<JRUN_HOME>\lib\ejipt_jms_client.jar  
<JRUN_HOME>\lib\ext\ejb.jar  
<JRUN_HOME>\lib\ext\jms.jar  
<JRUN_HOME>\lib\ext\jdbc.jar  
<JRUN_HOME>\lib\ext\jta.jar  
<JRUN_HOME>\lib\ext\jndi.jar
```
3. Click **OK** when you have finished adding all JAR files.
4. Right-click on the  testharnessProject and select  **Build....** The test harness must build successfully to continue with the EJB deployment.
5. Go to <JRUN\_HOME>\bin directory and stop the JRun 3.0 Admin/Default server by executing the following command in MS DOS command window

```
jrun -stop
```

**Note:** You *must* stop the JMC prior to attempting to execute the Test Harness or Web Application. Failure to do so prevents proper initialization of the JRun EJB Engine.

## Starting the JRun EJB Engine

You will have to start the JRun EJB engine in order to execute the EJB application. Since the command contains a number of switches and a path statement, you may want to create a short batch file to assist in starting the engine.






1. Access a command prompt, type this command, and then press Enter.

```
java -Dejpt.home="<drive-name>\Allaire\JRun" -
Djava.security.policy="c:\Program Files\Allaire\JRun\lib\jrun.policy"
-classpath
"<JRUN_HOME>\lib\ejpt.jar;<JRUN_HOME>\lib\ext\ejb.jar;<JRUN_HOME>\li
b\ext\jms.jar;<JRUN_HOME>\lib\ext\jndi.jar;<JRUN_HOME>\lib\ext\jta.ja
r;<JRUN_HOME>\lib\ext\parser.jar;<JRUN_HOME>\lib\ext\jdbc.jar;<drive-
name>\<JDBC driver class file or directory" allaire.ejpt.Ejpt
```

### Example

```
java -Dejpt.home="d:\Allaire\JRun" -
Djava.security.policy="c:\Program Files\Allaire\JRun\lib\jrun.policy"
-classpath
"d:\Allaire\JRun\lib\ejpt.jar;d:\Allaire\JRun\lib\ext\ejb.jar;d:\All
aire\JRun\lib\ext\jms.jar;d:\Allaire\JRun\lib\ext\jndi.jar;d:\Allaire
\JRun\lib\ext\jta.jar;d:\Allaire\JRun\lib\ext\parser.jar;d:\Allaire\J
Run\lib\ext\jdbc.jar;c:\inet" allaire.ejpt.Ejpt
```

In this example, *inet* is the MSSQL JDBC directory. The example loads the EJB, *default.RoleHome*, *default.LoginSessionHome* and *default.UserHome*. The JRun EJB engine is ready to execute the EJB application.

2. From the COOL:Joe 1.1 SP1 Project Tree, select the  testharnessProject and expand it.
3. Right-click on  <Component>Frame and then click  **Test...** This displays a small window that contains a **Task** menu from which you can select a method to test.
4. Click on **Task** and then click a method name to test it. The appropriate results display on the method panel. You can test  <Component>Frame.html and your generated  Web Application in a similar fashion.

## **Summary**

COOL:Joe 1.1 SP1 makes use of a plug-in architecture for application server support. This allows in-place additions to the deployment targets supported by COOL:Joe's *Deployment Wizard*. These additions make it possible for Computer Associates International, Inc. to make new deployment targets available between major releases of the COOL:Joe product software.

After performing the procedures outlined in this document, you should have successfully deployed your EJB to a JRun Server. You should also have been able to test your EJB using any or all of these objects generated by COOL:Joe 1.1SP1:

- Java application (standalone) Test Harness
- HTML/Servlet test harness
- Generated Web Application