

FuegoBPM Enterprise 5.5 with WebSphere 6.0 Installation Guide

Fuego, Inc.

FuegoBPM Enterprise 5.5 with WebSphere 6.0 Installation Guide

by Fuego, Inc.

Published January, 2005 - Version 5.5. Revision 10 - June, 2006.

Copyright © 2001-2006 Fuego, Inc.

FuegoBPM Enterprise 5.5 with WebSphere 6.0 Installation Guide

Copyright 2001-2006 Fuego, Inc. All rights reserved.

This documentation is subject to change without notice. This documentation and the software described in this document contains proprietary trade secrets and confidential information of Fuego, Inc. and is also protected by U.S. and other copyright laws and applicable international treaties. Use of this documentation and the software is subject to the license agreement between you and Fuego, Inc. If no such license agreement exists, you may not use this documentation and software in any manner whatsoever. Unauthorized use of the documentation or software, or any portion of it, will result in civil liability and/or criminal penalties. U.S. Patent Pending.

Fuego, Fuego 4, Component Manager, Process Designer, Work Portal, Orchestration Engine, Execution Console, Process Analyzer, Organization Administrator are trademarks or registered trademarks of Fuego, Inc.

FuegoBPM 5, FuegoBPM Studio, FuegoBPM Designer, FuegoBPM Enterprise Administration Center, FuegoBPM Work Portal, FuegoBPM Portal Console, FuegoBPM Archive Viewer, FuegoBPM Logviewer, FuegoBPM Express Server, FuegoBPM Enterprise Server, FuegoBPM Application Server Edition, FuegoBPM Web Console, FuegoBPM Process Analyzer, FuegoBPM Data Store, FuegoBPM Dashboard, FuegoBPM BAM, FuegoBPM Portlets, FuegoBPM Suite, FuegoBPM Deployer, FuegoBPM Failover, FuegoBPM VCS, FuegoBPM Ant Tasks, FuegoBPM FDI, FuegoBPM Help Viewer, FuegoBPM Server are trademarks or registered trademarks of Fuego, Inc.

InstallAnywhere is a registered trademark of Zero G Software, Inc. Solaris and Java are trademarks of Sun Microsystems, Inc. Windows is a registered trademark of Microsoft Corporation.

All other trademarks, trade names, and service marks are owned by their respective companies.

Table of Contents

1. Introduction	7
2. Minimal Requirements	8
3. Installing FuegoBPM Enterprise 5.5 for IBM WebSphere 6.0	10
Installation Steps	10
4. Configuring the FuegoBPM Directory Service	15
Creating FuegoBPM Directory Service	15
Configuring FuegoBPM Enterprise	15
Defining the Directory Services	16
Installing the drivers	17
Adding a Directory Service	18
Oracle as the Directory Service	19
Configuring FuegoBPM Web Application Server for Web Console deployment	25
Installing FuegoBPM Web Console	27
Installing FuegoBPM Web Console with no graphical interface	28
Starting FuegoBPM Web Console	31
5. Configuring a FuegoBPM Server for IBM's WebSphere Application Server	35
Login into the Web Console	35
Web Console Interface	35
Servers	37
Creating a Server	38
Creating a FuegoBPM Server for WebSphere	40
Oracle Server Database	44
Creating a Server Database	48
FuegoBPM Server's Oracle Database	49
6. Configuring FuegoBPM Server Resources on IBM's WebSphere Application Server	51
J2EE application server files	51
Installing FuegoBPM Libraries EAR in WebSphere	53
Creating Fuego Directory Service Connection Pool and Data Source	59
Defining Oracle's Authentication Data	59

Creating FuegoBPM Directory Service Data Source	61
Creating FuegoBPM Server Data Source	65
Creating Queues and Topics for FuegoBPM Server for Automatic Activity execution and News	70
Creating a Bus	71
Creating a Queue Connection Factory for ToDoItems Queue	72
Creating a Queue Destination for the ToDoItems Queue	77
Creating the JMS Queue	78
Creating the JMS activation specification	80
Creating a Connection Factory for FuegoBPM Server News Topic	83
Defining the FuegoBPM Server Topic Destination	85
Restarting Server and Testing	88
7. Installing FuegoBPM J2EE Deployer, FuegoBPM Server and FuegoBPM Work Portal on an Application Server	89
FuegoBPM J2EE Deployer	89
FuegoBPM J2EE Deployer for WebSphere	90
Installing FuegoBPM J2EE Deployer into WebSphere	91
FuegoBPM Server EAR	98
FuegoBPM Work Portal Web Application EAR and WAR files	100
Installing FuegoBPM Work Portal Web Application	102
Installing FuegoBPM Portal Console Web Application	103
Publishing and Deploying Projects	104
Publishing a Project	105
Deploying a Project	114
FuegoBPM Project EAR	117
8. Additional steps	119
9. Logging and Tracing	120
10. Troubleshooting	121
11. Appendix A: Creating FuegoBPM Directory Service Database using SQL-Plus	132
Getting the schema SQL creation script	132
Running the schema SQL creation script in Oracle SQL-Plus	135
12. Appendix B: Creating FuegoBPM Server Database using SQL-Plus	139

Creating a Server Database using an SQL script	139
Running the Server DB SQL creation script in Oracle SQL-Plus	140
13. Appendix C: Tips	143
Tips when using Oracle 9i Database	143
14. Appendix D: Configuring Oracle's JDBC Provider in WebSphere	144
Configuring Oracle's JDBC Provider	144
Setting up Oracle's JDBC Driver Environment Variable ..	146
15. Appendix E: Undeploying and Redeploying a Project	149
Undeploying and Redeploying a Project in J2EE	149
16. Appendix F: Quick Cross Reference for Resources	151
Cross Reference for Resources in WebSphere	151

Chapter 1. Introduction

Introduction

The following document describes all the steps to install and configure FuegoBPM Enterprise 5.5 for IBM WebSphere 6.0. It, as well, explains how to deploy a FuegoBPM Project implemented in FuegoBPM Studio on the FuegoBPM Server deployed on IBM WebSphere's Application Server connecting to an Oracle 9i RDBMS for the FuegoBPM Server and FuegoBPM Directory Service. FuegoBPM Server will also be using WebSphere's Embedded Messaging infrastructure.

Chapter 2. Minimal Requirements

Minimal Requirements

FuegoBPM 5 Enterprise for WebSphere

FuegoBPM Enterprise for IBM WebSphere must be fully installed and the Administration Center running.

WebSphere Minimal Requirements

FuegoBPM 5.5 Enterprise for IBM WebSphere runs on IBM WebSphere 6.0.

If the FuegoBPM Applications run in a clustering environment the following minimal requirements should be taken into consideration:

- When using a load balancer to access the clustered FuegoBPM Work Portal, the load balancer must be configured with sticky sessions.

Oracle 9i Minimal Requirements

The Oracle 9i Database that FuegoBPM is connecting to needs to be prepared to execute XA Transactions. If the XA support is not provided with the default Oracle Instance installation, contact your Oracle's DBA.

FuegoBPM uses Oracle 10g JDBC drivers when connecting to Oracle 9i Database.

Fuego strongly recommends that the latest 10g JDBC driver is used (at least 10g). Fuego has found some erratic behavior when using

old versions of the driver specially when working with Oracle Cluster deployments.

JMS Provider Minimal Requirements

FuegoBPM Server needs to use a messaging infrastructure for the processing of process automatic activities as well as scheduled tasks since all these events are processed by a Message Driven Bean on the J2EE implementation.

FuegoBPM relies on the JMS 1.0.2.b Specification. WebSphere's Embedded Messaging infrastructure already comes bundled with WebSphere and it is prepared to participate in XA transactions. In the case another messaging infrastructure is used, WebSphere's Administrator should make sure the JMS Provider Drivers are capable of managing XA transactions and that its implementation supports JMS 1.0.2.b. As well, FuegoBPM is capable of running on JMS 1.1. FuegoBPM connects through JMS to a Queue and a Topic. Minimal requirements for these are:

Queue: It needs to be XA compliant and non-persistent.

Topic: It needs to be XA compliant and non-persistent.

Chapter 3. Installing FuegoBPM Enterprise 5.5 for IBM WebSphere 6.0

Installation Steps

Configuring the Fuego Directory Service

The first thing to do after the FuegoBPM 5.5 Enterprise for IBM WebSphere is successfully installed, is to proceed with the configuration and setup of the environment to run a FuegoBPM Server on the WebSphere's Application Server to execute Business Processes modeled on FuegoBPM Studio.

FuegoBPM Server for IBM WebSphere relies on a Directory Service to enforce Business Process Security as well as using it as a centralized repository for FuegoBPM Metadata.

Creating FuegoBPM Directory Service

The FuegoBPM Administrator needs to create the FuegoBPM Directory Service. In the example, the FuegoBPM Administrator deploys Fuego Directory Service on an Oracle 9i RDBMS. Analogous procedure should be followed when deploying FuegoBPM Directory Service on another RDBMS or LDAP compliant Directory Service.

FuegoBPM Enterprise 5.5 for IBM WebSphere provides a desktop application for creating FuegoBPM Directory Service. This application is **FuegoBPM Administration Center**. This application may be launched using the shortcuts created at installation time or in background mode. Using this application, the administrator can create the FuegoBPM Directory Service, as well as, install the corresponding drivers for the database.

Configuring FuegoBPM Web Application Server for Web Console deployment

After successfully creating the FuegoBPM Directory Service, if you are in a Windows environment, it is convenient that the Admin

Service is installed as a Windows Service. The Admin Service starts a Tomcat embedded Web Server with FuegoBPM Web Console deployed on it. Installing it as a Windows Service enforces the FuegoBPM Web Console automatically started.

It is important to remark that the Web Console will be used to generate the EAR files to deploy on IBM's WebSphere.

Configuring a FuegoBPM Server for IBM's WebSphere Application Server

The next step in the setup is to configure the FuegoBPM Server. This configuration is achieved by using FuegoBPM Web Console. Steps to create and configure the FuegoBPM Server include:

- Create the FuegoBPM Server
- Creating the FuegoBPM Server backend RDBMS

Configuring FuegoBPM Server Resources on IBM's WebSphere Application Server

This section outlines how to create the J2EE Resources needed by the FuegoBPM Server to be deployed in IBM's WebSphere Application Server. The objective is to use all existing services provided by IBM's WebSphere Application Server for centralized resource administration. These resources include creating JDBC Data Sources as well as Topics and Queues for FuegoBPM Server internal asynchronous communication.

- Copying and configuring FuegoBPM Libraries (EAR) into WebSphere Application Server
- Creating FuegoBPM Directory Service Connection Pool and Data Source

- Setting up Oracle's JDBC Driver Environment Variable
 - Configuring Oracle's JDBC Provider
 - Defining Oracle's Authentication Data
 - Creating Fuego Directory Service Data Source
 - Creating Fuego Server Data Source
-
- Creating Queues and Topics for Fuego Server for Automatic Activity execution and News
 - Creating a Bus
 - Creating a Queue Connection Factory for ToDoItems Queue
 - Creating a Queue Destination for the ToDoItems Queue
 - Creating the JMS Queue
 - Creating the JMS activation specification
 - Creating a Connection Factory for Fuego Server News Topic
 - Defining the Fuego Server Topic Destination

Installing FuegoBPM Applications on IBM WebSphere

After the J2EE resources have been created in IBM's Application Server, FuegoBPM Administrator provides the EAR (Enterprise Application Files) files to the IBM WebSphere Administrator for installation.

Installing FuegoBPM J2EE Deployer

FuegoBPM provides the J2EE Deployer application, which can be used by FuegoBPM Administrator to deploy server and project applications from FuegoBPM Web Console. That way, manual deployment in WebSphere can be avoided for those applications.

FuegoBPM J2EE Deployer also simplifies the administration (start, stop, uninstall, etc) of server and project applications, by allowing FuegoBPM Administrator to execute those tasks directly from FuegoBPM Web Console.

Installing FuegoBPM J2EE Server

Once the FuegoBPM J2EE Deployer has been successfully deployed, you need to deploy FuegoBPM J2EE Server. The FuegoBPM J2EE Server is a collection of Services common to all FuegoBPM deployed projects. These services come in the form of an EAR file that is created within FuegoBPM Web Console by FuegoBPM Administrator.

FuegoBPM Administrator should generate the Server EAR (using the *Create EAR*) option, and then, through the FuegoBPM Deployer, install the server application in WebSphere.

Installing FuegoBPM Work Portal Web Application & FuegoBPM Portal Console Web Application

In J2EE a web application can be package as a WAR (Web Archive) as well as an EAR (Enterprise ARchive) file. FuegoBPM 5.5 provides Work Portal and Portal Console in both WAR and EAR formats and only one of them needs to be installed. The preferred one is the EAR file, specially because the J2EE Deployer is capable of managing it.

Note



having both WAR and EAR files installed may cause conflicts.

Installing FuegoBPM Project

FuegoBPM Administrator has the capability of publishing and deploying FuegoBPM Projects from the FuegoBPM Web Console. In addition, if the FuegoBPM Deployer Application is properly deployed

and configured, it enables FuegoBPM Administrator to deploy these projects into the Application Server.

The specific section explains how to publish and deploy a FuegoBPM Project until it is promoted to the Application Server.

Chapter 4. Configuring the FuegoBPM Directory Service

Creating FuegoBPM Directory Service

Configuring FuegoBPM Enterprise

Launch the **Administration Center**.

Click the option **Configuration** in the main window of the FuegoBPM Enterprise Administration Center and start configuring your FuegoBPM Enterprise.

▶ Start Web Application Server

■ Stop Web Application Server

■ Launch FuegoBPM™ Web Console

■ Launch FuegoBPM™ Work Portal

■ Launch FuegoBPM™ Portal Console

■ Launch FuegoBPM™ Archive Viewer

 **Configuration**

✕ Exit

When you select this option, the **Configuration** window displays. It is organized with tabs to administer and configure properties for:

- Directory
- Web Server Application
- FuegoBPM Web Console
- FuegoBPM Work Portal
- FuegoBPM Portal Console
- Service Pack Updates

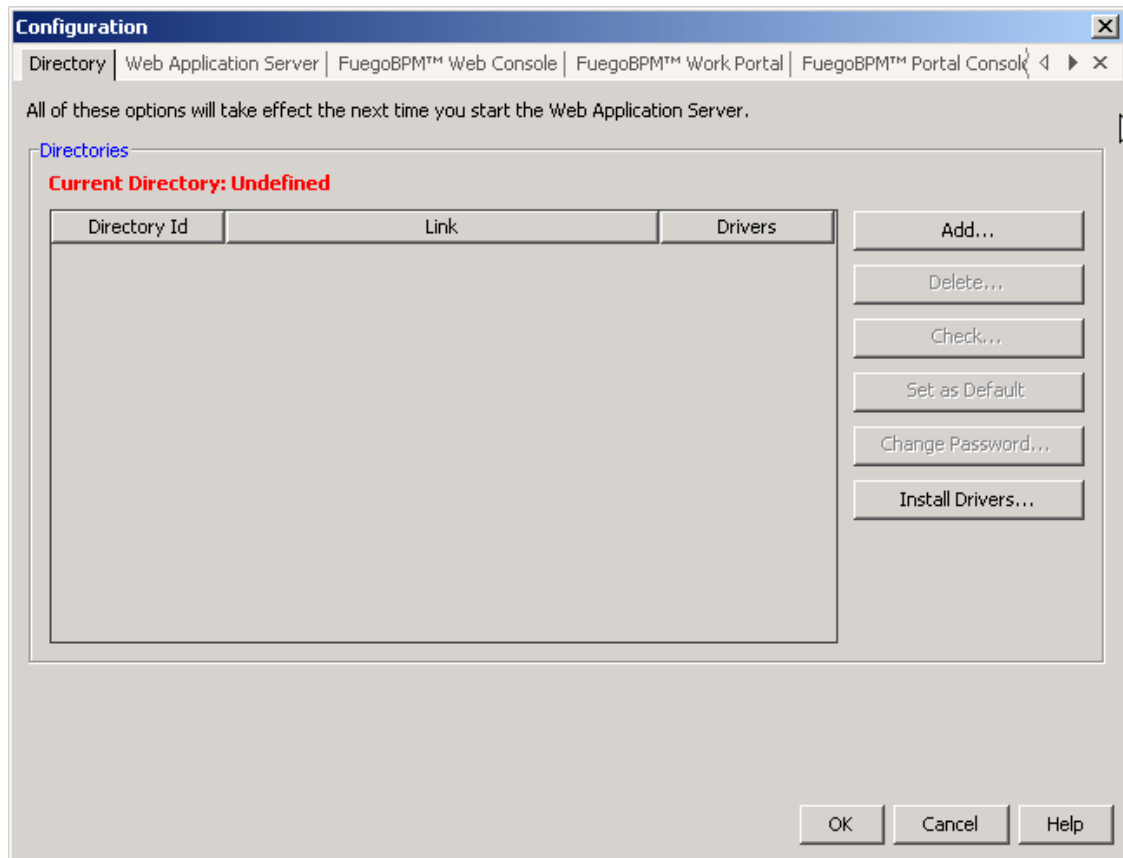
Note



You can also configure the FuegoBPM Enterprise without using the FuegoBPM Administration Center. To perform this configuration and Directory management you can use some Ant tasks. Refer to the System Administration Guide for detailed steps.

Defining the Directory Services

Defining the directory service is the first step in your FuegoBPM Enterprise implementation. To define a directory service, you must first install the corresponding drivers and then add the required directory.



Note



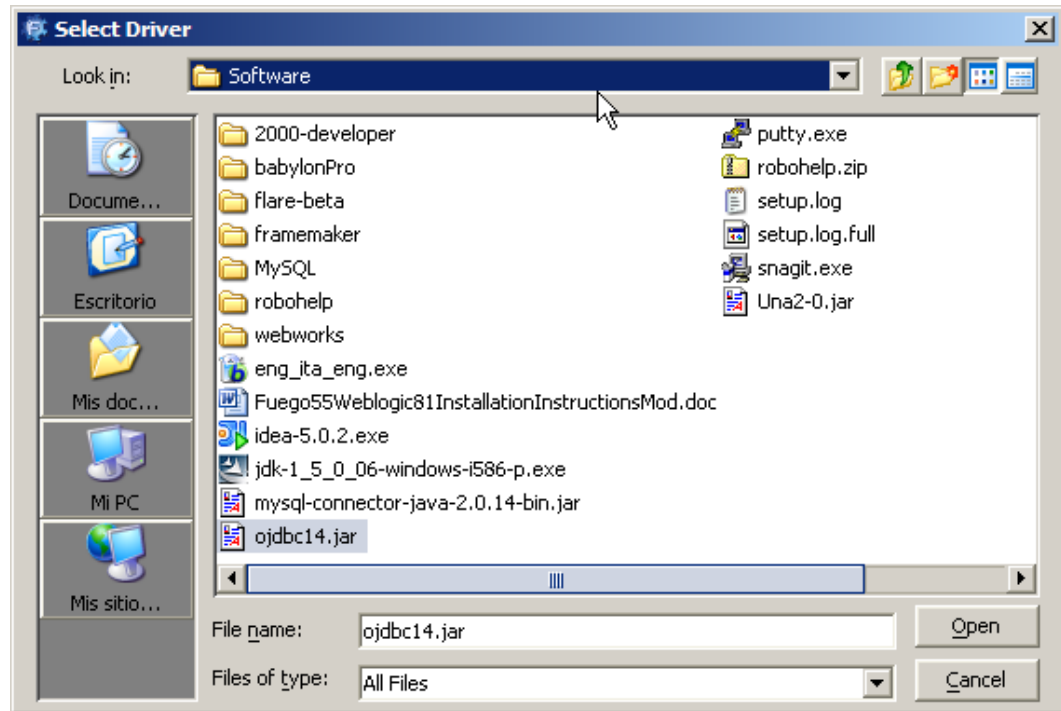
You must restart the Web Application Server for your changes to take place.

Installing the drivers

If you are using a RDBMS for the Directory Service you must install the correct driver BEFORE creating the Directory Service and logging into the RDBMS.


To install a driver

1. Select the button **Install Drivers....**
2. Browse to locate the jar file and click Open.



3. Click Ok to end installing the driver.

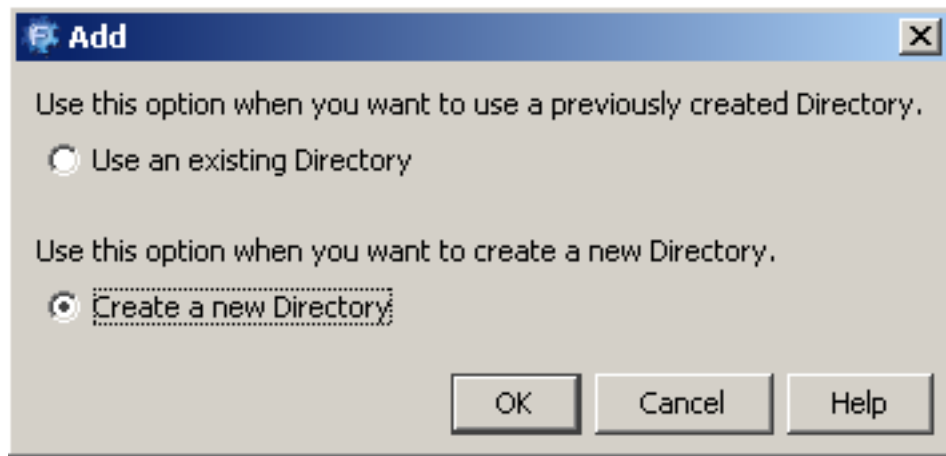
Note

 the driver to install is Oracle 10g JDBC driver that connects to the Oracle 9i RDBMS.

Adding a Directory Service

To add a new directory service,

1. Go to the **Directory** tab and click the **Add** button. The *Add* dialog box appears.
2. The following dialog will be presented to decide whether a new Directory Service should be created or not. Select one of the options. Click Ok.



3. A wizard opens to guide you through the creation of the directory.

Oracle as the Directory Service

FuegoBPM Enterprise Administration Center provides access to an Oracle database so that it can be used as a directory service. The database is accessed via a JDBC interface.

To configure the Oracle database

1. Click the Configuration link on the FuegoBPM Enterprise Administration Center main window. The Configuration dialog box appears.
2. Click the Directory tab.
3. Click the **Add** button and select to *create a new directory or use an existing one* and click **OK**.
4. The wizard to create a directory is opened. Give a name to the directory you are about to create in the **Directory Id** field. This name is the name that later will be used within the FuegoBPM system to identify this directory service configuration.

5. Select the **Provider**. In this case, choose the option *oracle*. Complete the fields in the tabs displayed in the dialog as explained below.

Basic Configuration

1. **Show the SQL sentences** - It should be checked if the Administrator wants to get the SQL Script executed to create the Directory Service Database structure. Refer to section Getting the schema SQL creation script in the System Administrator Guide to learn how to use this feature.
2. **Database host** - The host of the database server where the FuegoBPM FDI will be deployed.
3. **Database port** - The port for the Oracle server where the FDI will be deployed. Oracle JDBC driver connects with Oracle 9i using TCP/IP's protocol.
4. **Organization logical name** - The logical name of the organization to which the FDI will belong.
5. **Administrator User** - Oracle's Administrator user ID. This is a user with enough permission to create a Schema, a Login and tables on the created Directory Service Database.
6. **Administrator Password** - Administrator password.
7. **Schema name** - The name of the owner of the schema to create in the Oracle instance that will contain the FDI tables. This name is used to create the Oracle User that will contain the FDI tables. This Oracle Schema or User will be created when executing the SQL Script provided when the Wizard is started.
8. **Schema password** - The password of the owner of the schema to create. The password for the Schema Name specified in the

previous field.

9. **Confirm password** - Retype the password for confirmation. It is just for consistency when providing passwords as its value is not visualized to the Administrator entering the password.
10. **SID** - Schema ID.
11. **Specify JDBC url** - Check this box if you want to specify a URL different than the default one (connecting to a single Oracle Instance). Mainly if you are working with an Oracle clustering environment rack.
12. **JDBC url**: Define the custom Oracle JDBC URL when working with clustering.

Directory ID: DocDir

Provider: Oracle JDBC

Basic | Advanced

Show SQL Statements: ☐

Database Host: sputnik

Database Port: 1521

Organization Logical Name: company

Administrator User: system

Administrator Password: *****

Schema Name: docdir

Schema Password: *****

Confirm Password: *****

SID: sputnik

Specify JDBC URL: ☐

JDBC URL:

Start Cancel

Advanced Configuration

1. **Administrator ID** - The Administrator ID *root* is the Oracle user through which FuegoBPM will access the Oracle database, the

Fuego Directory Service Administrator's ID. All FuegoBPM participants will access the Oracle database using the *root* user. If FuegoBPM handles the authentication, Root will be the first participant created in FuegoBPM (as defined above).

2. **Administrator password** - Administrator *root* password.
3. **Confirm password** - Retype the password for confirmation.
4. **Fuego Handles authentication** - If each participant with FuegoBPM will not be an Oracle user, then select *Fuego handles authentication*. This is the preferred authentication mechanism since all Fuego Business Process participants will be persisted in the Direcotry Serivce Database. If this checkbox is not selected, the authentication will rely on Oracle's database authentication. Foor more information refer to the Oracle FDI instructions.
5. **Table space** - The Fuego Administrator can decide in which tablespace the tables for the FDI tables will reside..
6. **Temporary tablespace** - The Fuego Administrator can decide in which tablespace the temporary structures related to the FDI Users should reside..
7. **Profile** - The Fuego Administrator can associate an Oracle profile to the FDI User bein created..

The screenshot shows a configuration window for the FuegoBPM Directory Service. It has a title bar and a main content area. At the top, there is a 'Directory ID' field with the value 'DocDir'. Below this is a 'Provider' dropdown menu set to 'Oracle JDBC'. There are two tabs: 'Basic' (selected) and 'Advanced'. Under the 'Basic' tab, there are several fields: 'Administrator ID' with the value 'root', 'Administrator Password' with the value '****', a checked checkbox for 'Fuego handles authentication', and three empty text fields for 'Tablespace', 'Temporary Tablespace', and 'Profile'. At the bottom right of the window are 'Start' and 'Cancel' buttons.

Directory ID	DocDir
Provider	Oracle JDBC
Basic Advanced	
Administrator ID	root
Administrator Password	****
Fuego handles authentication	<input checked="" type="checkbox"/>
Tablespace	
Temporary Tablespace	
Profile	
Start Cancel	

After successfully creating the Database, Oracle's Administrator should grant permissions to the FuegoBPM Directory Service Oracle User to access the tables:

- DBA_2PC_PENDING
- DBA_2PC_NEIGHBORS
- DBA_PENDING_TRANSACTIONS

This is needed in case transactions need to be rolled back when involving information persisted in the FuegoBPM Directory Service. Log in as "sys as sysdba". The following statement can be executed with Oracle's SQL Plus:

```
GRANT SELECT ON DBA_PENDING_TRANSACTIONS TO SCHEMA_NAME;
```

where SCHEMA_NAME is the Oracle schema name given when the Directory Service schema was created (e.g. *docdir* is the Schema Name).

Configuring FuegoBPM Web Application Server for Web Console deployment

Some options and preferences for the Web Application Server can be configured through the FuegoBPM Enterprise Administration Center from the *Web Application Server* tab of the Administration Center.

To configure the Web Application Server

1. Click the Configuration link on the FuegoBPM Enterprise Administration Center main window. The Configuration dialog box appears.
2. Click the Web Application Server tab.

Note



You must restart the Web Application Server for your changes to take place.

Preferences

Preferences that can be set are:

- **JDK home directory** - The absolute path where the JDK is located. By default, the directory where the JDK is located when FuegoBPM Enterprise is installed (e.g. c:\fuego5.5\j2ee\jre).
- **Port** - This is the port used by the Web Console embedded Tomcat. The default port is set to 8585.
- **Shutdown port** - This is the port used internally to stop the Web Console embedded Tomcat. The default shutdown port is set to 8885.
- **Start FuegoBPM Web Console on Web Application Server startup** : this property enables the **Launch FuegoBPM Web Console** option in the Admin Center. In the case the Work Portal is deployed in an application server this checkbox should remain unselected, as it will be deployed later with the Fuego Server deployment.
- **Start FuegoBPM Work Portal on Web Application Server startup**: this property enables the **Launch FuegoBPM Work Portal** option in the Admin Center. In the case the Work Portal is deployed in an application server this checkbox should remain unselected, as it will be deployed later with the Fuego Server deployment.
- **Start FuegoBPM Portal Console on Web Application Server startup**: click on this property to enable the **Launch FuegoBPM Portal Console** option in the Admin Center. In the case the Work Portal is deployed in an application server this checkbox should remain unselected, as it will be deployed later with the Fuego Server deployment.
- **Start FuegoBPM Archive Viewer on Web Application Server startup**: click on this property to enable the **Launch FuegoBPM Archive Viewer** option in the Admin Center. In the case the Work Portal is deployed in an application server this checkbox

should remain unselected, as it will be deployed later with the Fuego Server deployment.

After changing any of the values, click **OK** to save them.

If you select **Cancel** a confirmation dialog is displayed asking if you still want to save the changes.

- **Show confirmation when exiting and the Web Application Server is running:** check or not this box according to your preferences, if you want to ask for the exit confirmation if the WAS is running.
- **Service Name:** This is the name of the Service to be granted to the Fuego Web Console Service. You to provide a name different than the default one if you have to FuegoBPM installations in the same machine.

Installing FuegoBPM Web Console

You can install the **Web Application Server** as a Service.

The Web Application Server starts a Tomcat embedded with **FuegoBPM Web Console** deployed on it. Installing it as a service enforces the Web Console to automatically start.

The Web Console, as well, is used in an Application Server environment to generate the EAR files to deploy on the Application Server.

Install as a Service using a graphical platform

To install the Web Console as a service for Windows or Unix like systems (excluding Solaris):

1. Go to the **Web Application Server** tab, section *Install as Service*.
2. Change, if necessary the **Service Name**. This is the name by which the Web Console service is identified. By default it is *Fuego Web Console*. Customizing this name allows you to have two installations in the same machine.
3. Click the **Install as Service** button. This will install the Web Console as a service in your operating system. By doing this, you are no longer required to manually start the Web Console from the Administration Center.

To uninstall the Web Console as a service:

1. Click the **Uninstall as Service** button.

Note



To install the Web Console as a service in **Solaris**, refer to the System Administrator Guide.

Installing FuegoBPM Web Console with no graphical interface

Installing FuegoBPM Web Console as a Service with no graphical interface

If you have a graphical interface, you can install the Web Console as a service using the Admin Center.

If you do not have a graphical interface to install nor uninstall the Web Application Server as a service you can do it manually.

Windows platform

To **install** the Web Console as a service for **Windows**:

1. Go to the installation directory *\$FUEGO/enterprise/bin*.
2. Run the following script:

```
installwebconsole -name ProductionWebConsole
```

where *ProductionWebConsole* is the **Service name** for the Web Console service.

To **uninstall** the Web Console as a service for **Windows**:

1. Go to the installation directory *\$FUEGO/enterprise/bin*.
2. Run the following script:

```
uninstallwebconsole -name ProductionWebConsole
```

where *ProductionWebConsole* is the **Service name** for the Web Console service that is being uninstalled.

Unix like system (except Solaris)

To **install** the Web Console as a service for a **Unix like system** execute the following commands:

```
cp $FUEGO_HOME/bin/fuegoautostartEnterprise /etc/init.d  
chkconfig --add fuegoautostartEnterprise
```

To **uninstall** the Web Console as a service for a **Unix like system** execute the following commands:

```
rm /etc/init.d/fuegoautostartEnterprise  
chkconfig --del fuegoautostartEnterprise
```

The *fuegoautostartEnterprise* script executes the shell script `$FUEGO_HOME/bin/startwebconsole.sh` when receiving *start* as argument, or `$FUEGO_HOME/bin/stopwebconsole.sh` when receiving *stop* as argument.

Solaris system

To **install** the Web Console as a service for **Solaris system** execute the following commands:

```
cp $FUEGO_HOME/bin/startwebconsole.sh /etc/init.d/startwebconsole  
cp $FUEGO_HOME/bin/stopwebconsole.sh /etc/init.d/stopwebconsole  
cd /etc/init.d  
chmod 744 startwebconsole stopwebconsole  
chown root:sys startwebconsole stopwebconsole  
ln startwebconsole /etc/rc3.d/Sstartwebconsole  
ln stopwebconsole /etc/rc3.d/Kstopwebconsole
```

To **uninstall** the Web Console as a service for **Solaris system** execute the following commands:

```
rm /etc/init.d/startwebconsole  
rm /etc/init.d/stopwebconsole  
rm /etc/rc3.d/Sstartwebconsole  
rm /etc/rc3.d/Kstopwebconsole
```

Starting FuegoBPM Web Console

To start Web Console, you can:

1. Start Web Console from the Admin Center. In this case the service is started automatically,
2. Start the installed Web Console service from the command line,
3. Start Web Console not as a service, or
4. Have it already started if it was installed as a service and during the system startup, the WebConsole service was correctly started.

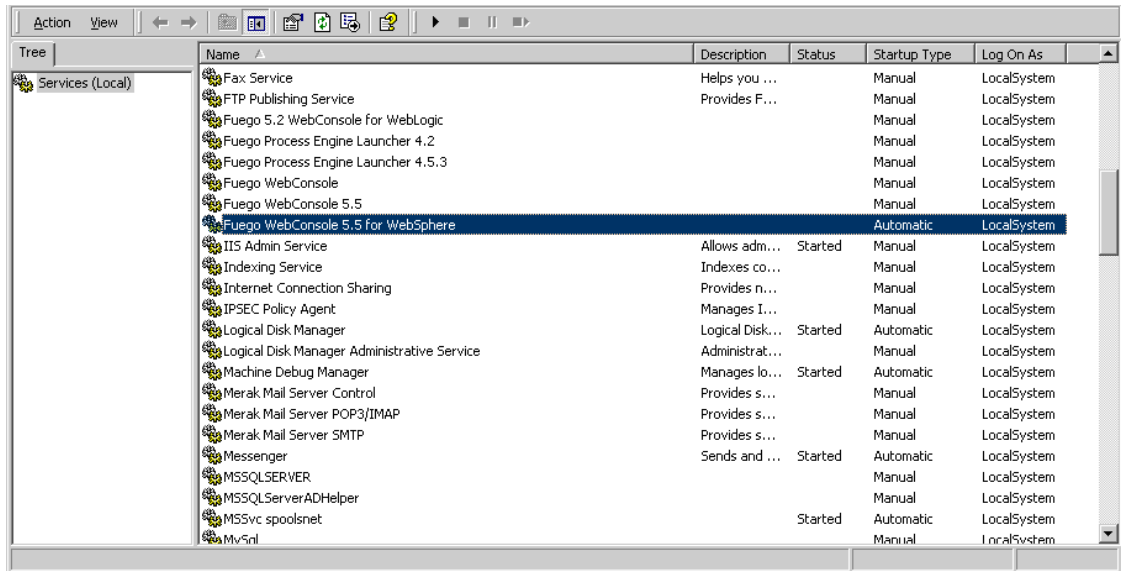
Starting Web Console from the Admin Center

To start Web Console:

1. Click on the **Start Web Application Server** link in the main panel of the FuegoBPM Administration Center.

2. Once the Web Application Server is started, click on **Launch FuegoBPM Web Console**.

Check the Web Console Service was properly started by checking the system Services Panel depicted in the figure below.



Starting the installed Web Console Service from the command line

To **start** the Web Console Service for **Unix like systems (except Solaris)** run the following:

```
. $FUEGO_HOME/bin/fuegoautostartEnterprise start
```

To **stop** the Web Console Service for **Unix like systems (except Solaris)** run the following:

```
. $FUEGO_HOME/bin/fuegoautostartEnterprise stop
```


If you have changed the *Service Name* when installing as a service the script name is `fuegoautostartEnterprise_serviceName`.

To **start** the Web Console Service for **Solaris** run the following:

```
startwebconsole
```

To **stop** the Web Console Service for **Unix like systems (except Solaris)** run the following:

```
stopwebconsole
```

Starting the Web Console not as a service

To **start** Web Console for **Windows** run the following:

```
$FUEGO_HOME/bin/startwebconsole.bat
```

To **stop** Web Console for **Windows** run the following:

```
$FUEGO_HOME/bin/stopwebconsole.bat
```

To **start** Web Console for **Unix like systems (including Solaris)** run the following:

```
$FUEGO_HOME/bin/startwebconsole.sh
```

To **stop** Web Console for **Unix like systems (including Solaris)**
run the following:

```
. $FUEGO_HOME/bin/stopwebconsole.sh
```

Chapter 5. Configuring a FuegoBPM Server for IBM's WebSphere Application Server

Login into the Web Console

The FuegoBPMo Administrator should use a browser to connect to the FuegoBPM Web Console as illustrated below. The URL to connect to the Web Console is **http://host:port/webconsole**.

- *host* is the machine in which the Web Console Windows Service has been installed and,
- *port* is the TCP/IP port configured for the Web Console.

Complete *User Name* with the *Administrator user ID* provided when creating FuegoBPM Directory Service and its password in the *Password* field to log into FuegoBPM Web Console. Click on the *Login* button to proceed with the login.

Web Console Interface

This interface enables FuegoBPM Administrators to create and maintain the proper environment for FuegoBPM Suite applications.

Configuring a FuegoBPM Server for IBM's WebSphere Application Server



There are some general characteristics and considerations that you should take into account:

- Although the Web Console is mainly accessed through the FuegoBPM Enterprise Administration Center application, you can open the Web Console using any browser with the appropriate network settings. However, the Web Console Server must be running to access the interface using a browser (for more information, refer to the *FuegoBPM Enterprise Administration Center* section of this System Administration Guide).
- The FuegoBPM Web Console has two main panels. The left panel contains the web console tree. Users can click any item listed on the web console tree to display in the right panel information and available tasks associated with the item.
- Users can refresh and exit the application using the **Reload** and **Logout** options, which are in the right upper corner of the browser.

- The elements associated with a specific item (FuegoBPM Server under the **Servers** option, for instance) are displayed in a list. Each element in the list has a check box associated with it. The tasks that allow multiple elements (deleting several participants, for example) are performed for all items with check boxes selected. You can work with either one element or a group of elements. Make sure you make your selections *before* you press the task button.
- Two additional options are present in all those screens of the application where a list of elements appear:
 - **Filter:** Allows a user to set a filter. Only elements that meet the filter criteria will be shown in the list. For example all the participants whose *Display Name* begins with **A** in the **Participants** pane.
 - **Preferences:** Enables the user to configure the maximum amount of rows displayed in each list.
- You can set the web application language by configuring the internet options of your browser.

Note



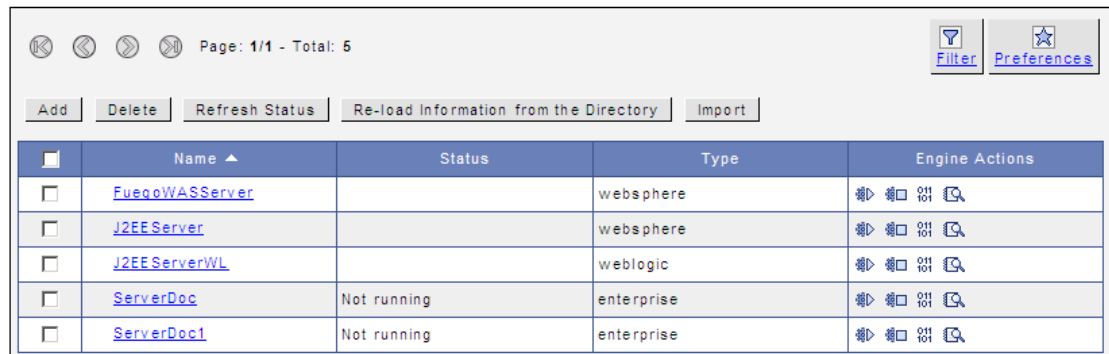
Using your browser's *Back* and *Next* buttons to change pages is not recommended. Instead, use the web console tree (from the left panel) to navigate through the application.

Servers

If you select the servers option in the left pane, a new window

appears on the right side of the browser with a list of all existing FuegoBPM Servers. From this pane, users can administer the Servers. It is possible to start and stop Servers as well as to view the startup log when an error occurs during the startup process.

Servers



	Name ▲	Status	Type	Engine Actions
<input type="checkbox"/>	FuegoWASServer		websphere	
<input type="checkbox"/>	J2EEServer		websphere	
<input type="checkbox"/>	J2EEServerWL		weblogic	
<input type="checkbox"/>	ServerDoc	Not running	enterprise	
<input type="checkbox"/>	ServerDoc1	Not running	enterprise	

The icons that appear next to each server included in the list allow you to start/stop the server, to launch a thread dump and to view the startup log.

When you operate with a J2EE Server, the start/stop buttons use the FuegoBPM Deployer in order to start and stop the application. The status of the server is also obtained from the deployer. There are three possible states: **application not installed** (in this case, you have to install the application from the J2EE application server files generation page before being able to start or stop the server), **application stopped** (if it's not running), or **application started**.

If the Deployer is not configured nor running, these operation cannot be performed and consequently the status is not displayed.

Creating a Server

You must create at least one FuegoBPM Server to run and manage processes you plan to publish and deploy. The FuegoBPM Server is responsible for the following:

- Accepting requests generated, for example, from the Work Portal
- Executing required tasks
- Maintaining the state of all instances flowing through the deployed processes

Typically, Servers are created by the FuegoBPM system administrator who is familiar with network connections, proxies, ports, and so on. Some of the things you must know to create a Server include the following:

- The relational database type and connection information
- Familiarity with process design and the number of instances that will flow through a process in a period of time
- The amount of concurrent users that are going to be connected to the application

To create a Server a wizard guides you through the required steps depending on the FuegoBPM Server type, or J2EE based type.

If the server is J2EE based, the wizard creates the configurations to access the server database and to access the FDI database. However, these configurations are not visible in the **Configurations** category of the Web Console.

The Server database access configuration can be edited through the **Basic Configuration** tab, clicking the *Edit server database configuration* link.

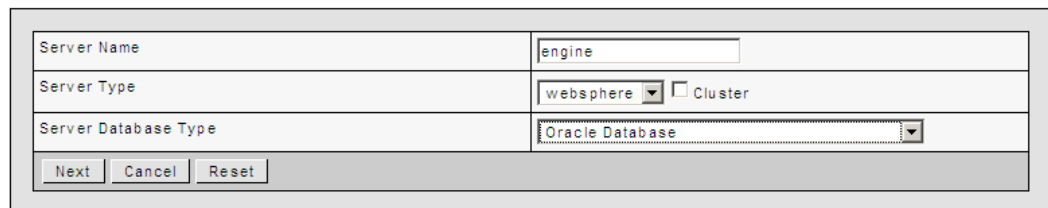
The remote configurations to access the server runtime database and the FDI database through the Application server can be visualized and changed if required in the **Application Server** tab.

Creating a FuegoBPM Server for WebSphere

To create a FuegoBPM Server for WebSphere

1. Click on Servers entry in the menu tree. The list of existing servers appears.
2. Click **Add**. The first step of the creation wizard *Choose Server Type* is displayed.

[Servers](#) > Choose a Server type.



The screenshot shows a dialog box titled "Choose a Server type." with the following fields and controls:

Server Name	engine
Server Type	websphere <input type="checkbox"/> Cluster
Server Database Type	Oracle Database
Next Cancel Reset	

3. Enter the **Name** of the Server.
4. Select the **Server Type** accordingly. In this case as "websphere".
5. The **cluster** check box is available when your license includes clustering. If you do not select it, the cluster specific options are not shown later.
6. Select the **Server database type**. In this case Oracle is used as the target backend RDBMS for business process instance persistence.
7. Click **Next** to continue.
8. The *Edit Configuration* screen displays. Configure the data for the enterprise server database configuration access. For details, see Server Database Considerations. Click **Next** to continue.

Configuring a FuegoBPM Server for IBM's WebSphere Application Server

[Servers](#) > [Choose a Server type.](#) > Edit Configuration Server 'engine' database configuration .

Type	
Name	Server 'engine' database configuration
Type	SQL Database
Subtype	Oracle Database

Properties	
Host	sputnik
Port	1521
SID	sputnik
User	engineDB
Password	*****
Confirm Password	*****
Schema	engineDB
Driver Type	thin
<input type="checkbox"/> Database String	

Advanced	
Tablespace	
Temporary Tablespace	
Profile	

Next Cancel Reset


9. The **J2EE Info** screen displays to visualize the configuration for the Application Server, and the FuegoBPM Deployer. The following panel shows the names for IBM's WebSphere resources used internally by the FuegoBPM Server deployed on WebSphere. These resources may be modified if IBM's WebSphere Administrator already has naming conventions. IBM's Administrator will be in charge of creating these resources after successfully completing the FuegoBPM Server creation. In the **Listener bindings for message-driven beans** property, you can select *Listener Port* or *Activation specification JNDI name*. The first option should be used for WebSphere 5.x version (the proposed name for the Listener Port is *ToDoQueueListenerPort*). If WebSphere 6.x version is used, FuegoBPM can use activation specification (as WS 6.x supports J2EE 1.4 specification) for Message Driven Beans communication. In that case, select this option in the

Listener combo. Change the default name for the activation specification

[Servers](#) > [Choose a Server type.](#) > [Edit Configuration Server 'engine' database configuration.](#) > J2EE Info

Summary of the configuration needed in your websphere application Server	
Server Runtime Datasource Lookup Name	<input type="text" value="XAengineDS"/>
Server FDI Datasource Lookup Name	<input type="text" value="XAFDIDS"/>
JMS Queue Name	<input type="text" value="queue/ToDoQueue"/>
JMS Queue Connection Factory	<input type="text" value="XAConnectionFactory"/>
JMS Topic Name for Server News	<input type="text" value="topic/EngineNews"/>
JMS Topic Connection Factory	<input type="text" value="XATopicConnectionFactory"/>
Listener bindings for message-driven beans	Activation Specification JNDI Name <input type="text" value="EngineActivSpec"/>

Note

 The values given to these attributes **must** exactly map the Websphere's Resources naming.

The following box shows properties for FuegoBPM Deployer application that should also be configured. It is important to replace the `< server >` tag in the Fuego Deployer URL Text Field with the name of the host where the IBM's WebSphere Application Server is deployed. The same applies for the `< port >` tag that represents WebSphere's default http incoming port. By default in WebSphere, it is 9080.

FuegoBPM™ deployer for WebSphere configuration	
FuegoBPM™ Deployer URL	<input type="text" value="http://apolo 9080/fuego/deployer/servlet/worker"/>
WebSphere Server/Cluster Name (Ex.: server1)	<input type="text" value="server1"/>
Use FuegoBPM™ libraries smart detection	<input checked="" type="checkbox"/>
FuegoBPM™ libraries are present at WebSphere lib/ext directory	<input checked="" type="checkbox"/>
Use default server path for FuegoBPM™ libraries	<input checked="" type="checkbox"/>
Server Path for FuegoBPM™ Libraries	<input type="text"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/> <input type="button" value="Reset"/>	

- a. **WebSphere server/cluster name:** the name of the WS server (the default name is `server1`) or cluster.

- b. **Use FuegoBPM™ libraries smart detection:** Check this option if you want to rely on Fuego libraries detection. The recommend action is to leave this option checked.
- c. **FuegoBPM™ libraries are present at WebSphere lib/ext directory:** If smart detection is not selected, this property sets the websphere lib/ext as source directories for Fuego libraries.
- d. **Use default server path for FuegoBPM™ libraries:** Check this option if default path for Fuego libraries will be used (generally *WAS_HOME/Fuego-Websphere-Deployer*). Only available when previous option is not selected.
- e. **Server path for FuegoBPM™ libraries:** If the previous options are not selected, you can set the path for Fuego libraries. If selected, the recommendation is to use absolute path.

10. The next step sets the **Basic Configuration**.

11. Enter the **Home directory** path for the Server. By defining a home directory, an administrator can determine where to place resources that are generated exclusively for this Server.

12. Enter the path for the directory where the log is stored in the **Log directory**. The log keeps track of the events defined in the Log settings tab. By default, the FuegoBPM Studio installation creates a log directory under the enterprise entry that is the most frequently configured path. It is important to make the Log directories to point to valid and existing paths in the WebSphere file system host.

13. Click **Save**.

FuegoBPM Application Server Edition Database

Important consideration

The clock of the J2EE Application Server where the FuegoBPM Application Server edition is deployed and the clock of the database manager of the FuegoBPM Application Server MUST be synchronized. This applies also to cluster application servers. Each node clock of the Application Server must be synchronized with the one of the database.

If these applications were not synchronized the automatic execution items or the due execution items may be executed after the time they should. Besides, in a cluster Application Server, the audit trail of an instance may show the events in a wrong order, for example: "Arrive to New York" before "Started the car engine to go to New York".

Each time that the FuegoBPM Server starts, it verifies that synchronization to be correct. If it weren't, a SEVERE log item would be logged with the following description: **The host clock is not synchronized with the db host clock.**

Oracle Server Database

Oracle Server Database Configuration

The following software must be in place before you start the Server with an Oracle database:

- Oracle database
- Oracle TNSListener, ready to work remotely using a TCP/IP port
- Oracle JDBC driver

- Thin JDBC driver
- OCI JDBC driver

Each FuegoBPM Server must have a user that stores all process information when a process is published, deployed, and running, and information on the instances flowing through the process.

The Oracle TNSListener serves as a bridge to access the Oracle database. By default, the TNSListener is installed on TCP port 1521. This port is used in the configuration for each of the FuegoBPM Servers so that they can easily access the Oracle database.

Oracle JDBC drivers enable the Server to access the database. The driver also makes information persistent in the database.

The Oracle database settings are as follows:

- **Schema** - Schema is automatically created when a Server is added to the Web Console. The name cannot be changed.
- **Host** - The host name of the machine or server where Oracle resides.
- **Port** - Default port number for Oracle.
- **SID** - System identification for database; also used to connect to database. Sometimes called Oracle ID.
- **User** - The user name set up by the Oracle system administrator. (Automatically created when the Server is created.) Each Server is related to an Oracle user. The user name in the Oracle instance is preceded by the prefix *fdb_* and ends with the name of the Server. The user name is defined in the Web Console when the Server is created. Once defined, the name cannot be changed because other FuegoBPM Suite applications reference it.

- **Password** - The administrative password for the Oracle database.
- **Driver type** - The type of drive that Oracle uses (thin or oci8).
- **Database String** - Check this box if you want to specify a special JDBC URL. In the next box, define the JDBC URL. This may be useful when connecting to an Oracle RAC. For example: jdbc:oracle:thin@host.
- **Advanced** - See below.

On the Advanced tab

- **Tablespace** - Some database administrators divide databases into tablespaces to control and maintain table sizes. If your company uses tablespace names, enter the appropriate name here. Leave the field blank if there are no tablespaces and a default tablespace will be created. When the user name is created in the Execution Console, the user creation statement reference tablespaces. Consequently, it is necessary to define the tablespaces.
- **Temporary Tablespace** - Enter the appropriate temporary tablespace name here. This field is going to be used by FuegoBPM Server's database to perform temporary indexing for some access. TEMP of type TEMPORARY. This tablespace performs temporary operations for the Server.
- **Administrator user** - Enter the administrator user name for the Oracle database. This is the user name that will be used to create the Server's database user in the Oracle instance. The password will be required at user creation time.
- **Profile** - A profile is a set of limits on database resources. If you assign the profile to a user, that user cannot exceed the established limits in the profile. This allows the administrator to

limit the actions of a particular Oracle user. The Oracle administrator may have different profiles set for different groups of users so that there is control over what each group is authorized to use and over which resources from the database a particular group will have.

- **Use Timestamp for Date columns** - When you select this property it makes the DATE fields work like TIMESTAMP. If the property is not selected, the DATE SQL fields type store only the day and the TIMESTAMP field type store both, day and time.

User Permissions

When the Oracle user is created in the Execution Console, the following permissions are granted:

- CONNECT
- RESOURCE

Defaults Assigned to Users

When the Oracle user is created in the Execution Console, the following defaults are assigned:

- TEMPORARY TABLESPACE "TEMP"
- PROFILE "DEFAULT"

Default for Tables Inside an Oracle User

By default, when the Oracle user is created for a Server in the Web Console, the tables related to this user are created as follows:

- Table's default tablespace is assigned when the user is created. This tablespace is configured in the Database tab when configuring the database for the server.
- Storage characteristics are the default table characteristics.
- An optional cluster.
- Partitioning definitions.
- Index-organized or heap-organized.

Length of Transactions

Most of the transactions performed by the Server or Oracle are short in duration. Thus, it is not necessary to include large rollback segments. High concurrence over the Oracle database provides enough space to manage multiple transactions.

Oracle Instance Character Set

The Server does not require a specific character set. The character set used depends on the region or country where you reside.

Oracle Instance Language

The language chosen for Oracle is left to your discretion.

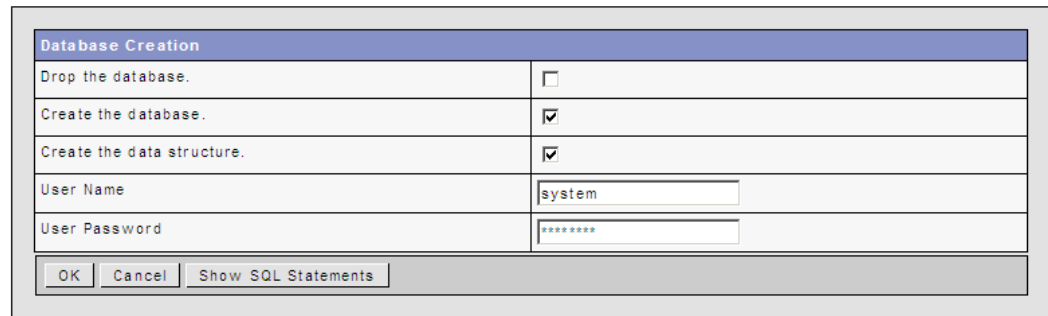
Creating a Server Database

After creating the FuegoBPM Server settings, the FuegoBPM Administrator should proceed to create the backend RDBMS for business process persistence. This is achieved by clicking on the **Manage database** link on the bottom of the Server Basic Configurations panel.

To create the Server database

1. Select the **Manage Database** option. The Database management pane appears.

[Servers](#) > [Edit Server engine.](#) > [Manage Database](#)



Database Creation	
Drop the database.	<input type="checkbox"/>
Create the database.	<input checked="" type="checkbox"/>
Create the data structure.	<input checked="" type="checkbox"/>
User Name	<input type="text" value="system"/>
User Password	<input type="password" value="*****"/>

2. Select the **Create database** and **Create data structure** check boxes.
3. Type a **User Name** with sufficient privileges (usually an administrator user like *system*), type the **User Password**, and click **OK**.

A message appears indicating that the database and the structure data have been properly generated.

By clicking the button **Show SQL Statements**, it is possible to download a script with the SQL statements that are executed during the database creation. Some database administrators might prefer executing them manually.

FuegoBPM Server's Oracle Database

If the engine is a J2EE based one, after successfully creating the Database, Oracle's Administrator should grant permissions to the FuegoBPM Server Oracle User to access the tables:

- DBA_2PC_PENDING
- DBA_2PC_NEIGHBORS

- DBA_PENDING_TRANSACTIONS

This is needed in case transactions need to be rolled back when the FuegoBPM Server cannot successfully complete a process activity task. Log in as *sys as sysdba*. The following statement can be executed with Oracle's SQL Plus:

```
GRANT SELECT ON DBA_PENDING_TRANSACTIONS TO SCHEMA_NAME;
```

where SCHEMA_NAME is the Oracle schema name given when the Server was created (e.g. *engineDB* is the Schema Name).

Chapter 6. Configuring FuegoBPM Server Resources on IBM's WebSphere Application Server

Configuring FuegoBPM Server Resources on IBM's WebSphere Application Server

This section outlines how to create the J2EE Resources needed by the FuegoBPM Server to be deployed in IBM's WebSphere Application Server. The objective is to use all existing services provided by IBM's WebSphere Application Server for centralized resource administration. These resources include creating JDBC Data Sources as well as JMS Topics and Queues for FuegoBPM Server internal asynchronous communication.

J2EE application server files


Both, the FuegoBPM Library JAR and the Cluster libraries, can be downloaded by clicking the file name. These libraries are not required to be generated as they are distributed with the installation of the product. Click the file name and browse the location to store it.

These files are also available at:

```
$FUEGO$/j2ee/j2ee/common/lib/fuegoj2ee-lib-all.jar
```

For clustering:

```
$FUEGO$/j2ee/j2ee/common/lib/cluster/fuegoj2ee-engine-cluster.jar  
$FUEGO$/j2ee/j2ee/common/lib/cluster/jgroups-core.jar
```

To generate the Server EAR, Portal EAR/WAR, Portal Administrator EAR/WAR and deployed projects EARs, click the  file generation icon that is next to these applications.

The Project EARs option, generates an EAR file per each project version, including the deployed processes in the current server.

If the FuegoBPM Deployer is not installed, or if it's not properly configured, a warning will be issued stating that there is no additional information to be shown, and the page will only display the icons to generate the files, and the links to download them. In this case, the installation of the EARs must be done manually.

[Servers](#) > [Edit Server engine.](#) > J2EE Application Server Files (EARs, WARs)

FuegoBPM™ Application Libraries

FuegoBPM™ Library JAR

This JAR file includes all FuegoBPM™ helper classes needed to run FuegoBPM™ project EARs in your application Server. It should be installed as a shared library.

[fuegoj2ee-lib-all.jar](#) 28.23 Mb May 2, 2006 7:19:08 PM

FuegoBPM™ Server EAR

The FuegoBPM™ Server EAR contains all the Server related classes to manage and execute the business services according to rules defined on the published business processes.

[01-eng-engine.ear](#) 9.15 Kb May 11, 2006 5:11:53 PM

FuegoBPM™ Work Portal WAR

These files contain the FuegoBPM™ Work Portal Web Application. This Web Application enables end users to participate in deployed business processes according to their assigned roles. It can be deployed in your application Server or on any Java Servlet compliant engine. The Web Application URI can be edited before creating the deployment file. The Web Application is available in WAR (Web ARchive) and EAR (Enterprise ARchive) file formats. Both archives are equivalent and only one of them needs to be installed.

[05-portal-XAFDIDS-websphere.war](#) 6.79 Mb May 11, 2006 5:12:29 PM

[05-portal-XAFDIDS-websphere.ear](#) 6.47 Mb May 11, 2006 5:12:35 PM

FuegoBPM™ Portal Console WAR

These files contain the Web Application designed to give process developers and FuegoBPM™ system administrators the ability to customize FuegoBPM™ Work Portal views and functionality. It can be deployed in your application Server or on any Java Servlet compliant engine. The Web Application URI can be edited before creating the deployment file. The Web Application is available in WAR (Web ARchive) and EAR (Enterprise ARchive) file formats. Both archives are equivalent and only one of them needs to be installed.

[06-portaladmin-XAFDIDS-websphere.war](#) 15.56 Mb May 11, 2006 5:12:53 PM

[06-portaladmin-XAFDIDS-websphere.ear](#) 15.55 Mb May 11, 2006 5:13:03 PM

Project EARs

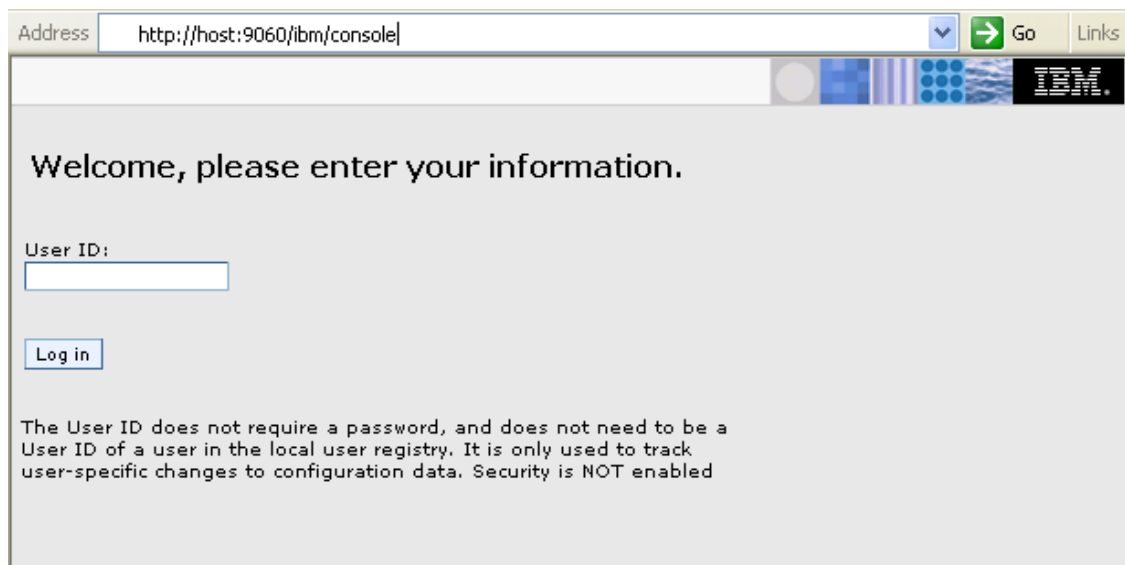
The FuegoBPM™ Deployer is not available. The functionality of the FuegoBPM™ Deployer includes the ability to install the J2EE Server and projects in your J2EE Application Server as well as to monitor the state of those applications.

On the other hand, if the FuegoBPM Deployer is installed and properly configured, this page will show more information, and will allow the user to install, uninstall, start and stop the applications.

Installing FuegoBPM Libraries EAR in WebSphere

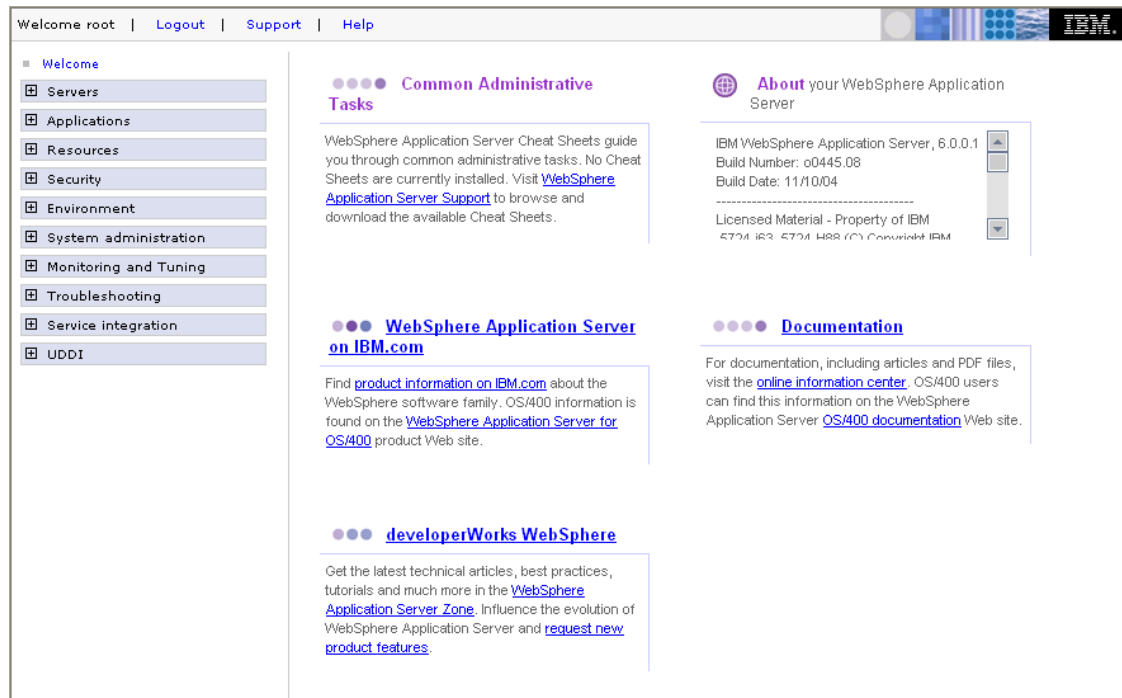
Login into IBM's Websphere Application Server Console

Login using WebSphere's Administrator user. By default, IBM's WebSphere Console runs on the following URL *http://host:9060/ibm/console*.



The screenshot shows a web browser window with the address bar containing `http://host:9060/ibm/console`. The page has an IBM logo in the top right corner. The main content area displays the text "Welcome, please enter your information." Below this, there is a "User ID:" label followed by a text input field. A "Log in" button is positioned below the input field. At the bottom of the page, a note states: "The User ID does not require a password, and does not need to be a User ID of a user in the local user registry. It is only used to track user-specific changes to configuration data. Security is NOT enabled".

Click on the *Log In* button to proceed. Once successfully logged in, you will be presented with IBM's WebSphere Application Server Configuration as depicted below.



Installing the Libraries

Installing FuegoBPM Libraries for WebSphere implies the creation of a shared library and the classloader configuration in each server that contains FuegoBPM application.

The FuegoBPM for J2EE uses EJBs as part of its implementation. In J2EE the client of EJBs needs a Stub class for the beans. FuegoBPM provides the pre-generated stubs classes at `$FUEGO$/j2ee/j2ee/websphere/was6stubs/fuegoj2ee-was6-stubs.jar`

Place the Fuego **lib jars** (`fuegoj2ee-lib-all.jar` and `fuegoj2ee-was6-stubs.jar`) in a file system that can be accessible from WebSphere .

1. Go to *Environment / Shared Libraries*
2. Add a new one as shown below (On a multinode environment this task should be done in each node where a FuegoBPM application is running).

[Shared Libraries](#) > **New**

Specifies a container-wide shared library that can be used by deployed applications.

Configuration

General Properties

* **Name**
BSMServer Libs

Description
Libraries

* **Classpath**
.commom\lib\fuegoj2ee-lib-all.jar
.commom\lib\fuegoj2ee-was6-stubs.jar

Native Library Path

Apply OK Reset Cancel

If the installation is a **cluster configuration**, two more **jars** must be added. They can be found in the *J2EE application server files (EARs, WARs)* page from *Server configuration* within the Web Console, or under `$FUEGO$/j2ee/j2ee/common/lib/cluster`.

Every time you update or modify the configuration, the WebSphere console suggests to *Save* the change. You can save every change as you make it or save all of them at the end.

[Shared Libraries](#) > **New**

Specifies a container-wide shared library that can be used by deployed applications.

Configuration

General Properties

* **Name**
BSMServer Libs

Description
Libraries

* **Classpath**
lib\fuegoj2ee-lib-all.jar
lib\fuegoj2ee-was6-stubs.jar
lib\cluster\fuegoj2ee-engine-cluster.jar
lib\cluster\jgroups-core.jar

Native Library Path

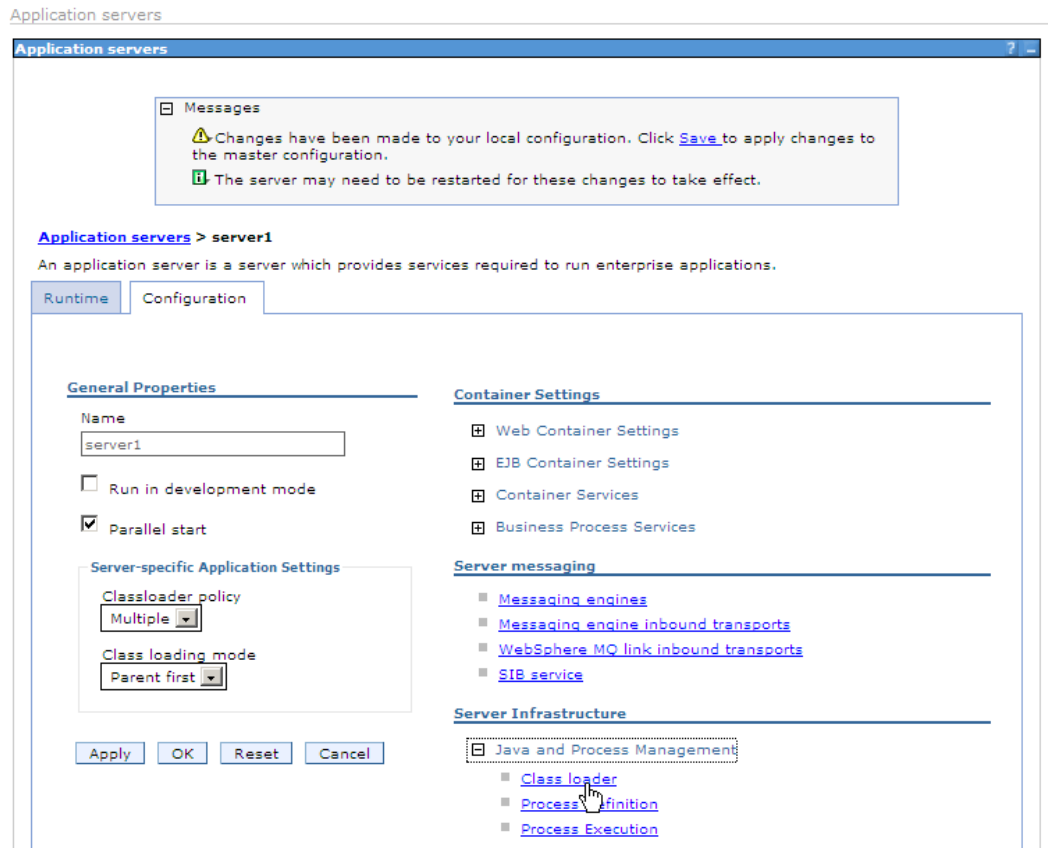
Apply OK Reset Cancel

Classloader configuration

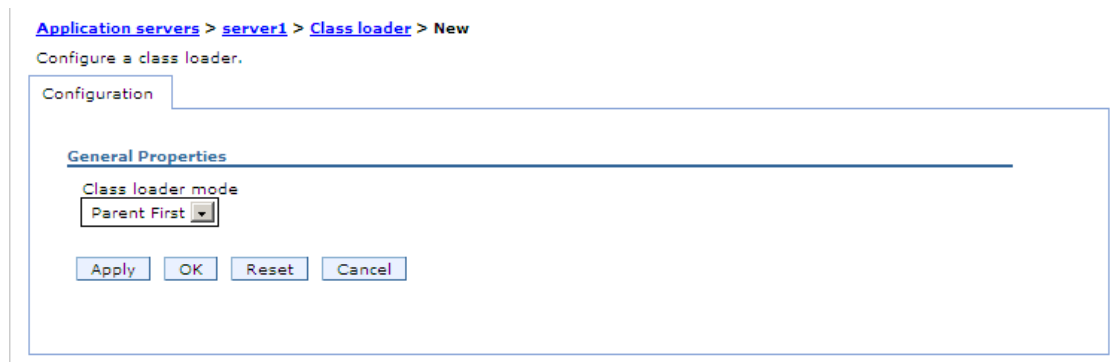
Then, a classloader configuration should be added or edited for each WS server running FuegoBPM applications.

1. Open the *server properties* page, as shown below:

Configuring FuegoBPM Server Resources on IBM's WebSphere



2. Select *Java and Process Management* -> *Class loader*.
3. Press *New* to add a new class loader configuration.



Enter the following:

- **Parent First** option.

After confirming the new classloader, open it by clicking on the *id* link.

[Application servers](#) > [server1](#) > Class loader

Configure a class loader.

⊞ Preferences

<input type="button" value="New"/> <input type="button" value="Delete"/>		
Select	Class loader ID ↕	Class loader mode ↕
<input type="checkbox"/>	ClassLoader_1147380742781	Parent First
Total 1		

1. Go to *Libraries* option.
2. Add the FuegoBPM shared library as shown below.

[Application servers](#) > [server1](#) > [Class loader](#) > [ClassLoader_1147380742781](#) > [Library Reference](#) > [New](#)

A library reference specifies a shared library file used by this application. Define a library reference for each library file that your application uses.

Configuration

General Properties

Library name

Press Ok to add the classloader configuration (remember that Save must be selected in order to persist the changes).

Creating Fuego Directory Service

Connection Pool and Data Source

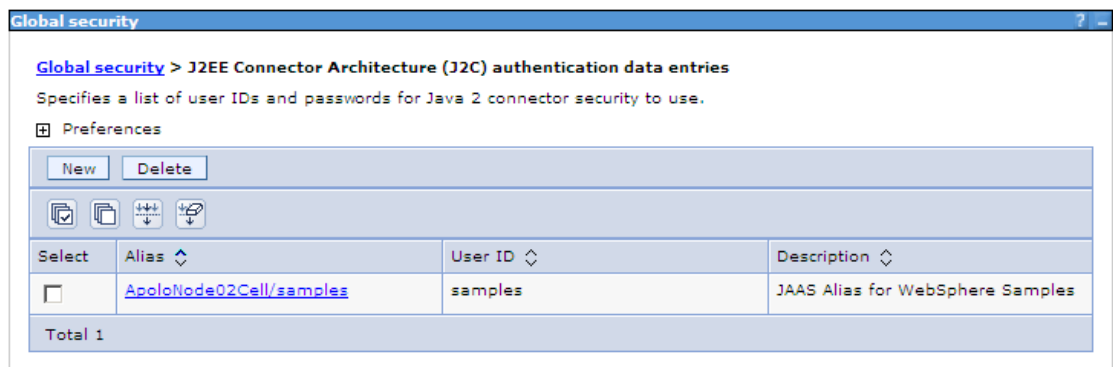
Defining Oracle's Authentication Data

Before proceeding to the next step, be sure that the **Oracle's JDBC Provider** is configured. If not, refer to the corresponding appendix.

The next step is to configure the authentication, both, for **FuegoBPM Directory Service** and **FuegoBPM Server Databases**.

To create the **Authentication Data**:

1. Go to *Security-> Global Security -> JAAS Configuration -> J2C Authentication Data*.



You need to create one entry for **FuegoBPM Directory Service** and another one for the **FuegoBPM Server**.

Click *New* to create first **FuegoBPM Directory Service Authentication Data**.

[Global security](#) > [J2EE Connector Architecture \(J2C\) authentication data entries](#) > **New**
Specifies a list of user IDs and passwords for Java 2 connector security to use.

Configuration

General Properties

* **Alias**

* **User ID**

* **Password**

Description

Enter or select the following:

- **Alias** : This is a logical name for the defined user/password.
- **User ID** : is the Oracle User defined when creating FuegoBPM Directory Service Schema name in the Admin Center.
- **Password**: contains the password for this Oracle User.

Click *OK* to save FuegoBPM Directory Service Authentication Data.

Likewise, an authentication data needs to be created for the **FuegoBPM Server backend Database**. These properties are depicted in the figure below.

[Global security](#) > [J2EE Connector Architecture \(J2C\) authentication data entries](#) > **New**

Specifies a list of user IDs and passwords for Java 2 connector security to use.

Configuration

General Properties

* Alias
EngineAuthData

* User ID
engineDB

* Password

Description
"engine" user f/AuthData

Apply OK Reset Cancel

- **User ID:** is the one defined when creating the Server as part of the *Edit Configuration*.

Click *Ok* to continue.

Creating FuegoBPM Directory Service Data Source

After creating FuegoBPM Directory Service Authentication Data, you need to create the **Data Source** to connect to the already created **FuegoBPM Directory Service Oracle Database**.

To create FuegoBPM Directory Service Data Source:

1. Go to *Resources -> JDBC Providers*.

JDBC providers

JDBC providers are used by the installed applications to access data from databases.





☐ Scope: Cell=**ApoloNode01Cell**, Node=**ApoloNode01**

☐ Cell : ApoloNode01Cell Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#)

→ ☒ Node : ApoloNode01

☐ Server : server1

Preferences

<input type="button" value="New"/> <input type="button" value="Delete"/>		
   		
Select	Name ↕	Description ↕
<input type="checkbox"/>	Oracle JDBC Driver (XA)	Oracle JDBC Driver (XA)
Total 1		

2. Click the Oracle XA JDBC Driver entry (e.g.: *Oracle JDBC Driver (XA)*). If you do not have an already Oracle JDBC Driver defined, follow the steps defined in *Appendix D: Configuring Oracle JDBC Provider in WebSphere*
3. Click the *Data Sources* link on the new rendered page to start creating the Data Source.

Click on the *New* button to open the configuration panel.

[JDBC providers](#) > [Oracle JDBC Driver \(XA\)](#) > [Data sources](#) > **New**

A data source is used by the application to access data from the database. A data source i supplies the specific JDBC driver implementation class.

Configuration

General Properties

* Scope

cells:ApoloNode02Cell:nodes:ApoloNodeDocArTeam

* Name

DirServ Oracle JDBC Driver XA DataSource

JNDI name

XAFDIDS



Use this Data Source in container managed persistence (CMP)

Description

DirServ New JDBC Datasource

Category

Configuring FuegoBPM Server Resources on IBM's WebSphere

Data store helper class name

☒ Select a data store helper class

Data store helper classes provided by WebSphere Application Server

Oracle9i and prior data store helper
(com.ibm.websphere.rsadapter.OracleDataStoreHelper)

Oracle10g data store helper
(com.ibm.websphere.rsadapter.Oracle10gDataStoreHelper)

☐ Specify a user-defined data store helper

Enter a package-qualified data store helper class name

Component-managed authentication alias

Component-managed authentication alias

ApoloNodeDocArTeam/DirServAuthData

Authentication alias for XA recovery

☐ Use component-managed authentication alias

☒ Specify:

ApoloNodeDocArTeam/DirServAuthData

Container-managed authentication

Container-managed authentication alias (deprecated in V6.0, use resource reference authentication settings instead)

ApoloNodeDocArTeam/DirServAuthData

Mapping-configuration alias (deprecated in V6.0, use resource reference authentication settings instead)

DefaultPrincipalMapping

Oracle data source properties

* URL

jdbc:oracle:thin:@sputnik:1521:sputnik

Apply OK Reset Cancel

- **Name:** This is a logical name to identify FuegoBPM Directory

Service Data Source.

- **JNDIName:** The field should match the value specified for the *Server FDI datasource lookup name* when creating the FuegoBPM Server for WebSphere in FuegoBPM Web Console.
- **Description:** This is a description for the FDI Data Source.
- **Component-managed Authentication Alias:** WebSphere's Administrator should pick the Authentication Data created for FuegoBPM Directory Service (e.g.: DirServAuthData).
- **Container-managed Authentication Alias:** WebSphere's Administrator should pick the Authentication Data created for FuegoBPM Directory Service (e.g.: DirServAuthData).
- **Mapping-Configuration Alias:** WebSphere's Administrator should select the entry *DefaultPrincipalMapping* from the dropdown.
- **URL:** set the URL to enable proper JDBC connectivity with the Oracle 9i Database: **jdbc:oracle:thin:@host:port:SID** (e.g. *jdbc:oracle:thin:@sputnik:1521:sputnik*). This URL value may change to adapt it to the host, port (default: 1521) and SID where Oracle is deployed as well as whether you are connecting to an Oracle 9i or superior Clustering environment.

Click *Ok* to continue. The recently created Data Source may be tested by clicking on the *Test Connection* button on the Data Source configuration panel. As well, this configuration may need a restart of the application server or the node agent when clustering, if the JDBC driver *ojdbc14.jar* was just installed.

Creating FuegoBPM Server Data Source

After creating FuegoBPM Server Authentication Data, you need to create the Data Source to connect to the already created FuegoBPM Server Oracle Database.

To create FuegoBPM Server Data Source:

1. Go to *Resources -> JDBC Providers*
2. Click on the Oracle XA JDBC Driver entry (e.g.: Oracle JDBC Driver (XA)). If you do not have an already Oracle JDBC Driver defined, follow the steps defined in *Appendix D: Configuring Oracle JDBC Provider in WebSphere*
3. Click on the *Data Sources* link on the new rendered page to start creating the Data Source.

Click on the *New* button to open the configuration panel.

[JDBC providers](#) > [Oracle JDBC Driver \(XA\)](#) > [Data sources](#) > **New**

A data source is used by the application to access data from the database. A data provider, which supplies the specific JDBC driver implementation class.

Configuration

General Properties

* Scope

cells:ApoloNode02Cell:nodes:ApoloNodeDocArTeam

* Name

Engine Oracle JDBC Driver XA DataSource

JNDI name

XAengineDS



Use this Data Source in container managed persistence (CMP)

Description

Engine New JDBC Datasource

Category

Data store helper class name

☒ Select a data store helper class

Data store helper classes provided by WebSphere Application Server

Oracle9i and prior data store helper (com.ibm.websphere.rsadapter.OracleDataStoreHelper)	▲
Oracle10g data store helper (com.ibm.websphere.rsadapter.Oracle10gDataStoreHelper)	▼

☐ Specify a user-defined data store helper

Enter a package-qualified data store helper class name

Component-managed authentication alias

Component-managed authentication alias

ApoloNodeDocArTeam/EngineAuthData ▼

Authentication alias for XA recovery

☐ Use component-managed authentication alias

☒ Specify:

ApoloNodeDocArTeam/EngineAuthData ▼

Container-managed authentication

Container-managed authentication alias (deprecated in V6.0, use resource reference authentication settings instead)

ApoloNodeDocArTeam/EngineAuthData

Mapping-configuration alias (deprecated in V6.0, use resource reference authentication settings instead)

DefaultPrincipalMapping

Oracle data source properties

* URL

jdbc:oracle:thin:@sputnik:1521:sputnik

Apply OK Reset Cancel

Enter or select the following:

- **Name:** This is a logical name to identify FuegoBPM Server Data Source.
- **JNDIName:** The field should match the value specified for the *Server Runtime Datasource Lookup Name* when creating the FuegoBPM Server for WebSphere in FuegoBPM Web Console.
- **Description:** This is the description for the FuegoBPM Server Data Source.
- **Component-managed Authentication Alias:** WebSphere's Administrator should pick the Authentication Data created for FuegoBPM Server (e.g.: EngineAuthData).
- **Container-managed Authentication Alias:** WebSphere's Administrator should pick the Authentication Data created for FuegoBPM Server (e.g.: EngineAuthData).
- **Mapping-Configuration Alias:** WebSphere's Administrator should select the entry *DefaultPrincipalMapping* from the dropdown.

- **URL:** set the URL to enable proper JDBC connectivity with the Oracle 9i Database: **jdbc:oracle:thin:@host:port:SID** (e.g. *jdbc:oracle:thin:@sputnik:1521:sputnik*). This URL value may change to adapt it to the host, port (default: 1521) and SID where Oracle is deployed as well as whether you are connecting to an Oracle 9i or superior Clustering environment.


Click *Ok* to continue. The recently created Data Source may be tested by clicking on the *Test Connection* button on the Data Source configuration panel. As well, this configuration may need a restart of the application server or the node agent when clustering, if the JDBC driver *ojdbc14.jar* was just installed.

After successful creation of both, the **FuegoBPM Directory Service** and **FuegoBPM Server Data Sources**, the **Data Source** panel displays as follows:

[JDBC providers](#) > [Oracle JDBC Driver \(XA\)](#) > [Data sources](#)

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

⊞ Preferences

New Delete Test connection Manage state...				
				
Select	Name ↕	JNDI name ↕	Description ↕	Category ↕
<input type="checkbox"/>	DirServ Oracle JDBC Driver XA DataSource	XAFDIDS	DirServ New JDBC Datasource	
<input type="checkbox"/>	Engine Oracle JDBC Driver XA DataSource	XAengineDS	Engine New JDBC Datasource	
Total 2				

Creating Queues and Topics for FuegoBPM Server for Automatic Activity execution and News

The following steps describe the configuration of Queues and Topics

for Automatic activities managed by the FuegoBPM Server.

Creating a Bus

Before creating the JMS Queues and Topics, the WebSphere Administrator should create a **Bus**.

To create a Bus:

1. Go to *Service Integration -> Buses*.
2. Click the *New* button.

The screenshot shows the 'Buses > New' configuration page in the WebSphere Administrator. The page has a blue header bar with the text 'Buses'. Below the header, there is a breadcrumb 'Buses > New' and a descriptive paragraph: 'A service integration bus supports applications using message-based and service-oriented architectures interconnected servers and clusters that have been added as members of the bus. Applications on the messaging engines associated with its bus members.' Below this is a 'Configuration' tab. Under the 'Configuration' tab, there are two sections: 'General Properties' and 'Topology'. The 'General Properties' section has two fields: '* Name' with the value 'TheBus' and 'UUID' with the value 'A18E00C6552659AD'. The 'Topology' section is on the right and contains a list of checkboxes: 'Bus members', 'Messaging engines', and 'Foreign buses'. A note on the right side of the 'General Properties' section states: 'The additional properties will not be a general properties for this item are sa'.

Click *OK*.

Bus members

To create a Bus member:

1. Select the created Bus
2. Within its *General Properties*, select the *Bus members* link.

[Buses](#) > [TheBus](#) > **Bus members**

Bus members are the servers and clusters that have been added to the bus.

☒ Preferences

☐ ☐ ☐ ☐

Select	Name
None	
Total 0	

Click the *Add* button and configure the server

Add a new bus member

Add a server or server cluster as a new member of the bus.

→ **Step 1: Select server or cluster**

Step 2: Confirm the addition of a new bus member

Select server or cluster

Choose the server or cluster to add to the bus

☒ **Server**

Server

Data store

☒ Default

Data source JNDI name

☐ **Cluster**

Cluster

Data store

* Data source JNDI name

Click *Next*.

Confirm the addition clicking on *Finish*

Creating a Queue Connection Factory for ToDoItems Queue

WebSphere's Administrator should proceed to create a JMS Connection Factory to successfully manage FuegoBPM Server **ToDoItems**. The *ToDoItems* is the list of pending items to be executed by the FuegoBPM Server. These items are the automatic activities as well as internal activities of the server.

For this exercise, WebSphere Embedded JMS Server is used. Other JMS configurations may be provided as appendixes to their document.

In order to create a Queue Connection Factory:

1. Go to *Resources->JMS Providers->Default messaging*.
2. Click on the *JMS Queue Connection Factory* link .
3. Click the *New* button in the new panel.

A new panel will be presented to complete the configuration of the *ToDoItems* Queue Connection Factory. The following panel depicts the needed parameters.

[Default messaging provider](#) > [JMS queue connection factory](#) > New

A JMS queue connection factory is used to create connections to the a messaging. Use queue connection factory administrative objects to m messaging provider.

Configuration**General Properties****Administration***** Scope***** Name***** JNDI name****Description****Category**

Connection

* Bus name

TheBus

Target

Target type

Bus member name

Target significance

Preferred

Target inbound transport chain

Provider endpoints

Connection proximity

Bus

Quality of Service

Nonpersistent message reliability

Express nonpersistent

Persistent message reliability

Reliable persistent

Advanced Messaging

Read ahead

Default

Temporary queue name prefix

Advanced Administrative

Component-managed authentication alias

(none)

☐ Log missing transaction contexts

☐ Manage cached handles

☐ Share data source with CMP

XA recovery authentication alias

(none)

Apply

OK

Reset

Cancel

- **Name:** This is a logical name for the *ToDoItems Queue Connection Factory*. In the example the name is *TheConnectionFactory*.

- **JNDIName:** This parameter should contain the value assigned to the *JMS Queue Connection Factory* Web Console's Server parameter.

The rest of the values should take default values.

Click *Ok* to save the Queue Connection Factory settings.

Creating a Queue Destination for the ToDoItems Queue

After creating the *ToDoItems Queue Connection Factory*, WebSphere's Administrator needs to create the Queue Destination.

In order to create a Queue Destination complete the following steps:

1. Select *Service Integration-> Buses*, in the Navigation pane on the left.
2. Select the created **Bus**
3. Select the *Destinations* link from the displayed window.
4. Select *New*.
5. Select **Queue** and click *Next*.

Enter or select the following:

- **Identifier:** enter a logical name for the Destination, then select *Next*.

Buses

Create new queue

Create a new queue for point-to-point messaging

→ Step 1: Set queue attributes

Step 2: Assign the queue to a bus member

Step 3: Confirm queue creation

Set queue attributes

Configure the attributes of your new queue

* Identifier
ItemsQDestination

Description

Next Cancel

- Select the bus member

Buses

Create new queue

Create a new queue for point-to-point messaging

Step 1: Set queue attributes

→ Step 2: Assign the queue to a bus member

Step 3: Confirm queue creation

Assign the queue to a bus member

Assign the queue to a bus member that will store and process the messages for the queue.

Bus member
ApoloNodeDocArTeam:server1

Previous Next Cancel

Select *Next*, and finally *Finish*.

Creating the JMS Queue

To create a JMS Queue:

1. Go to *Resources => JMS Providers => Default Messaging*.
2. Select the *JMS Queue* link
3. Click *New*.

Enter or select the following:

- **Name:** This is a logical name for the JMS Queue. In the example the name is *ToDoItemQueue*.
- **JNDI Name:** This parameter should contain the value assigned to the *JMS Queue Name* Web Console's Server parameter.
- **Bus name:** The bus name defined previously.
- **Queue name:** The Queue Destination defined for the bus (enabled once the bus is selected).
- Leave all other default values, select *OK*, and save your changes.

[Default messaging provider](#) > [JMS queue](#) > **New**

A JMS queue is used as a destination for point-to-point messaging. Use JMS queue destination administrative objects default messaging provider.

Configuration

General Properties

Administration

* Scope

cells:ApoloNode02Cell:nodes:ApoloNodeDocArTeam

* Name

ToDoItemQueue

* JNDI name

queue/ToDoQueue

Description

The screenshot shows a 'JMS Activation Specification' dialog box. It has two sections: 'Connection' and 'Advanced'. In the 'Connection' section, there are five fields: 'Queue name' (a dropdown menu with 'ItemsQDestination' selected), 'Bus name' (a dropdown menu with 'TheBus' selected), 'Delivery mode' (a dropdown menu with 'Application' selected), 'Time to live' (a text input field), and 'Priority' (a text input field). To the right of the 'Time to live' field is the label 'milliseconds'. In the 'Advanced' section, there is one field: 'Read ahead' (a dropdown menu with 'As connection factory' selected). At the bottom of the dialog are four buttons: 'Apply', 'OK', 'Reset', and 'Cancel'.

Creating the JMS activation specification

The **activation specification**, which is new for J2EE 1.4, has to be referenced from the FuegoBPM Server property *Listener bindings for message-driven beans*.

To create a JMS Activation Specification:

1. Go to *Resources->JMS Providers->Default messaging*
2. Select *JMS activation specification*
3. Click *New*

Enter or select the following:

- **Name:** This is a logical name for the JMS Activation Specification.
- **JNDI name:** This parameter should contain the value assigned to the Web Console's Server parameter for the *Listener bindings for*

message-driven beans when selecting *Activation Specification JNDI Name*.

- **Destination type:** *Queue* .
- **Destination JNDI name:** Destination defined for the bus.
- **Bus name:** Select the created bus.
- Leave all other default values

Select **OK**, and save your changes.

[Default messaging provider](#) > [JMS activation specification](#) > **EngineActivSpec**

A JMS activation specification is associated with one or more message-driven beans and p them to receive messages.

Configuration

General Properties

Administration

* Scope

cells:ApoloNode02Cell:nodes:ApoloNodeDocArTeam

* Name

EngineActivSpec

* JNDI name

EngineActivSpec

Destination

* Destination type

Queue ▾

* Destination JNDI name

queue/ToDoQueue

Message selector

Related Items

■ [J2EE Connec
authenticati](#)

■ [Buses](#)

Bus name

TheBus

Acknowledge mode

Auto-acknowledge

Target inbound transport chain

Additional

Authentication alias

(none)

Maximum batch size

Maximum concurrent endpoints

The image shows a configuration dialog box with two main sections: "Subscription Durability" and "Advanced".

Subscription Durability

- Subscription durability:** A dropdown menu with "Nondurable" selected.
- Subscription name:** An empty text input field.
- Client identifier:** An empty text input field.
- Durable subscription home:** An empty text input field.

Advanced

- Share durable subscriptions:** A dropdown menu with "In cluster" selected.
- ☐ **Share data source with CMP**
- Read ahead:** A dropdown menu with "Default" selected.

At the bottom of the dialog are four buttons: "Apply", "OK", "Reset", and "Cancel".

Creating a Connection Factory for FuegoBPM Server News Topic

IBM's WebSphere Administrator should also create a Topic so that the FuegoBPM Server can distribute News to other servers and connected clients.

To create a Connection Factory for News Topic:

1. Go to *Resources -> JMS Provider -> Default Messaging*.

2. Click on the *JMS topic connection factory* link.
3. Click on *New* to advance to the Topic Connection Factory configuration panel.

[Default messaging provider](#) > [JMS topic connection factory](#) > **New**

A JMS topic connection factory is used to create connections to the ass messaging. Use topic connection factory administrative objects to mar messaging provider.

Configuration

General Properties

Administration

* Scope

cells:ApoloNode02Cell:nodes:ApoloNodeDocArTeam

* Name

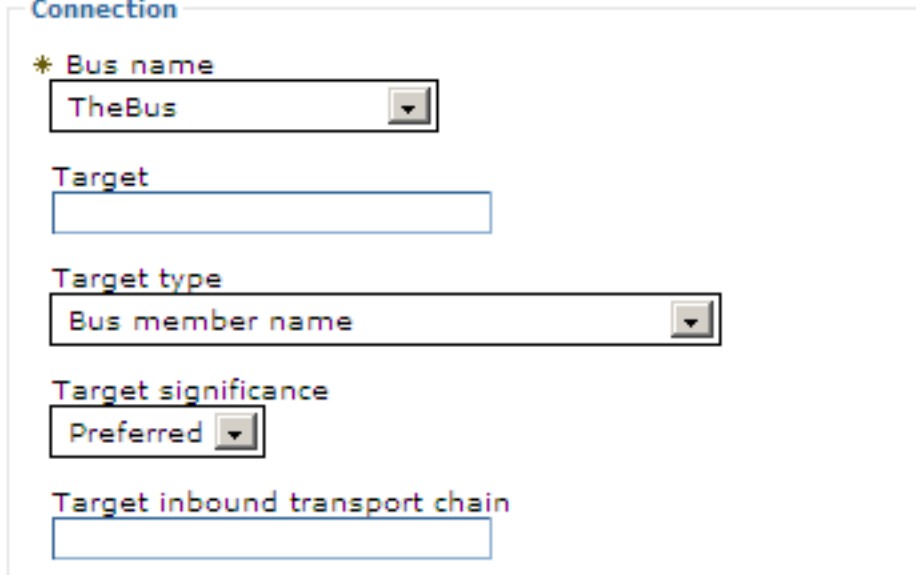
XATopicConnectionFactory

* JNDI name

XATopicConnectionFactory

Description

Category



The screenshot shows a 'Connection' configuration window. It has the following fields:

- Bus name:** A dropdown menu with 'TheBus' selected.
- Target:** An empty text input field.
- Target type:** A dropdown menu with 'Bus member name' selected.
- Target significance:** A dropdown menu with 'Preferred' selected.
- Target inbound transport chain:** An empty text input field.

Enter or select the following:

- **Name:** This is the name for the News Topic Connection Factory (e.g.: News Connection Topic).
- **JNDI Name:** This parameter should contain the value specified for the *JMS Topic Connection Factory* parameter for the FuegoBPM Server in FuegoBPM Web Console.
- **Bus Name:** Select the bus defined previously.

The rest of the parameters can take default values.

Click *Ok* to continue.

Defining the FuegoBPM Server Topic Destination

After successfully creating the Topic Connection Factory, create the Topic Destination.

1. Go to *Resources -> JMS Providers-> Default Messaging*.
2. Click on *JMS Topic*.
3. Click on *New*

[Default messaging provider](#) > **New**

A JMS topic is used as a destination for publish/subscribe messaging. Use topic destination JMS topics for the default messaging provider.

Configuration

General Properties

Administration

* Scope

cells:ApoloNode02Cell:nodes:ApoloNodeDocArTeam

* Name

EngineNews

* JNDI name

topic/EngineNews

Description

Connection

Topic name

Topic space

Bus name

JMS delivery mode

Time to live
 milliseconds

Message priority

Advanced

Read ahead

Enter or select the following:

- **Name:** This is the logical name for the News Topic Destination.
- **JNDI Name:** This parameter should contain the JNDI name specified for the *JMS Topic name for Server news* parameter for the FuegoBPM Server on the FuegoBPM Web Console.
- **Description:** This is a description for the News Topic.
- **Topic:** This is the name of the Topic Destination. By default, FuegoBPM names this Topic Destination *EngineNews*.

The rest should take default values.

Restarting Server and Testing

Once all resources have been defined:

1. *Save* the configuration.
2. Restart the application server.
3. Test the connection to the *Data Sources*.
 - a. Go to *Resources -> JDBC Providers*
 - b. Select *Oracle XA JDBC Driver entry*
 - c. Click the *Data Sources* link
 - d. Select both Data Sources, the FuegoBPM Directory Service and FuegoBPM Server
 - e. Click on the *Test Connection*

Chapter 7. Installing FuegoBPM J2EE Deployer, FuegoBPM Server and FuegoBPM Work Portal on an Application Server

Installing FuegoBPM J2EE Deployer, FuegoBPM Server and FuegoBPM Work Portal on an Application Server

Once the J2EE resources have been created in the Application Server, FuegoBPM Administrator should provide the EAR (Enterprise Application Files) files to the Application Server Administrator for installation.

The FuegoBPM J2EE Deployer application can be used by FuegoBPM Administrator to deploy server and project applications from FuegoBPM Web Console. That way, manual deployment in the Application Server can be avoided for those applications.

FuegoBPM J2EE Deployer also simplifies the administration (start, stop, uninstall, etc) of server and project applications, by allowing FuegoBPM Administrator to execute those tasks directly from FuegoBPM Web Console.

FuegoBPM J2EE Deployer

The FuegoBPM J2EE Deployer allows the management of Enterprise Applications (.ear files) in a J2EE Application Server.

The FuegoBPM Deployer is an Enterprise Application itself, that must be installed in the application server. There is a particular implementation for each vendor of supported application servers. Every implementation has a property in common: the deployer URL. The URL has the following format:

```
http://server:port/fuego/deployer/servlet/worker
```

where *server* is the host name of the machine where the Application Server is running and the *port* name is the port where the Application Server publish web applications (typically the same port where the web administrative console for the particular Application Server runs).

Here are some example urls:

```
http://localhost:8080/fuego/deployer/servlet/worker  
http://localhost:7001/fuego/deployer/servlet/worker  
http://localhost:9043/fuego/deployer/servlet/worker
```

By sending commands through this URL, the deployer is able to install, uninstall, start, stop, and check the status of installed applications. In the Web Console, it must be configured when creating a J2EE server or updated in the Application Server tab, and it is used when operating with J2EE servers or generating J2EE application server files.

It might be used through ant tasks as well.

FuegoBPM J2EE Deployer for WebSphere

FuegoBPM J2EE Deployer for WebSphere

The EAR file containing the FuegoBPM J2EE Deployer for WebSphere server side implementation must be installed in the Application Server. This ear file is called *wasj2eedeployer.ear* and is in the installation directory *j2ee/j2ee/websphere/deployer*. If it is a single server environment then it must be installed in the unique available server.

It can be deployed using the WebSphere Administrative Console. If the deployer is installed in a clustered environment, then the ear must be installed in the Server of the Deployment Manager Node (typically *dmgr*). The WebSphere Administrative Console does not support deploying ears in the Server of the Deployment Manager Node, so it must be deployed using other mechanism as using the *wsadmin.bat* or command line tools provided by WebSphere.

FuegoBPM WebSphere Deployer configuration parameters includes the standard url parameter and an additional parameter which is the *server/cluster* name. This parameter identifies the name for the server (or cluster for clustered servers) where FuegoBPM Servers and projects are going to be deployed. For example *server1* or *cluster1*.

Both parameters are specified at the FuegoBPM Web Console at the server creation time, or can be accessed in the Application Server properties page for the server.

Installing FuegoBPM J2EE Deployer into WebSphere

Installing FuegoBPM J2EE Deployer

IBM's Administrator should deploy this EAR through the following page (that can be found in the following WebSphere Console navigational tree path: *Applications -> Install New Application*). After clicking on the *Install New Application*, WebSphere's Administrator should complete the panel shown below.

Enterprise Applications

Preparing for the application installation

Specify the EAR, WAR or JAR module to upload and install.

Path to the new application.

☒ Local file system

Specify path

deployer\wasj2eedyeployer.ear

Browse...

☐ Remote file system

Specify path

Browse...

Context root

Used only for standalone Web modules (.war files)

Next

Cancel

With the help of the *Browse* button, localize the EAR mentioned above. Click *Next* to continue with the FuegoBPM J2EE Deployer Application deployment. Take the default values in the next panel as shown below and click *Next* to continue.

Enterprise Applications [Close page](#)

Preparing for the application installation ?

Choose to generate default bindings and mappings.

☐ Generate Default Bindings

Override:

☒ Do not override existing bindings

☐ Override existing bindings

Virtual Host

☒ Do not use default virtual host name for Web modules

☐ Use default virtual host name for Web modules:

Host name
default_host

Specific bindings file

Accept defaults and click *Next* to continue.

Enterprise Applications

Install New Application ?

Specify options for installing enterprise applications and modules.

→ **Step 1: Select installation options**

[Step 2: Map modules to servers](#)

[Step 3: Map virtual hosts for Web modules](#)

[Step 4: Summary](#)

Select installation options

Specify the various options that are available to prepare and install your application.

☐ Pre-compile JSP

Directory to install application

☒ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

Application name
Fuego WebSphere Deplc

☒ Create MBeans for resources

☐ Enable class reloading

Reload interval in seconds

☐ Deploy Web services

Validate Input off/warn/fail
warn

☐ Process embedded configuration

Installing FuegoBPM J2EE Deployer, FuegoBPM Server and FuegoBPM

Accept defaults and click *Next* to continue.

Enterprise Applications [Close page](#)

Install New Application

Specify options for installing enterprise applications and modules.

[Step 1 Select installation options](#)
→ **Step 2: Map modules to servers**
[Step 3 Map virtual hosts for Web modules](#)
[Step 4 Summary](#)

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the plug-in configuration file (plugin-cfg.xml) for each Web application which are routed through it.

Clusters and Servers:
WebSphere:cell=ApoloNode02Cell,node=ApoloNodeDocArTeam,server=server1 [Apply](#)

Select	Module	URI	Server
<input type="checkbox"/>	wasj2eedyployer.war	wasj2eedyployer.war,WEB-INF/web.xml	WebSphere:cell=ApoloNode02Cell,node=ApoloNodeDocArTeam,server=server1

[Previous](#) [Next](#) [Cancel](#)

Accept defaults and click *Next* to continue.

Enterprise Applications

Install New Application ?

Specify options for installing enterprise applications and modules.

[Step 1 Select installation options](#)
[Step 2 Map modules to servers](#)
→ **Step 3: Map virtual hosts for Web modules**
[Step 4 Summary](#)

Map virtual hosts for Web modules

Specify the virtual host where you want to install the Web modules contained in your application. You can install Web modules on the same virtual host or disperse them among several hosts.

☒ Apply Multiple Mappings

Select	Web module	Virtual host
<input type="checkbox"/>	wasj2eedyployer.war	default_host

[Previous](#) [Next](#) [Cancel](#)

The selected virtual host value defines the application port of FuegoBPM Deployer. By default, this value is 9080.

Accept defaults (if you agree with default virtual host selection) and click *Next* to continue.

Installing FuegoBPM J2EE Deployer, FuegoBPM Server and FuegoBPM

Enterprise Applications

Install New Application ? -

Specify options for installing enterprise applications and modules.

[Step 1](#) Select installation options

[Step 2](#) Map modules to servers

[Step 3](#) Map virtual hosts for Web modules

→ Step 4: Summary

Summary

Summary of installation options

Options	Values
Use Binary Configuration	No
Create MBeans for resources	Yes
Cell/Node/Server	Click here
Reload interval in seconds	
Enable class reloading	No
Process embedded configuration	No
Application name	Fuego WebSphere Deployer
Validate Input off/warn/fail	warn
Directory to install application	
Distribute application	Yes
Deploy Web services	No
Pre-compile JSP	No
Deploy enterprise beans	No

Previous

Finish

Cancel

Accept default values and click on the *Finish* button to finalize the FuegoBPM J2EE Deployer Application deployment.

Installing...

If there are enterprise beans in the application, the EJB deployment process can take several minutes. Please do not save the configuration until the process completes.

Check the SystemOut.log on the Deployment Manager or server where the application is deployed for specific information about the EJB deployment process as it occurs.

ADMA5016: Installation of Fuego WebSphere Deployer started.

ADMA5067: Resource validation for application Fuego WebSphere Deployer completed successfully.

ADMA5058: Application and module versions validated with versions of deployment targets.

ADMA5005: The application Fuego WebSphere Deployer is configured in the WebSphere Application Server repository.

ADMA5053: The library references for the installed optional package are created.

ADMA5005: The application Fuego WebSphere Deployer is configured in the WebSphere Application Server repository.

ADMA5001: The application binaries are saved in
C:\was6\IBMWebSphere\AppServer\profiles\DocArTeam\wstemp\1409340525\workspace\cells\ApoloNode02Cell\applications\Fuego
WebSphere Deployer.ear\Fuego WebSphere Deployer.ear

ADMA5005: The application Fuego WebSphere Deployer is configured in the WebSphere Application Server repository.

SECJ0400: Successfully updated the application Fuego WebSphere Deployer with the appContextIDForSecurity information.

ADMA5011: The cleanup of the temp directory for application Fuego WebSphere Deployer is complete.

ADMA5013: Application Fuego WebSphere Deployer installed successfully.

Application Fuego WebSphere Deployer installed successfully.

To start the application, first save changes to the master configuration.

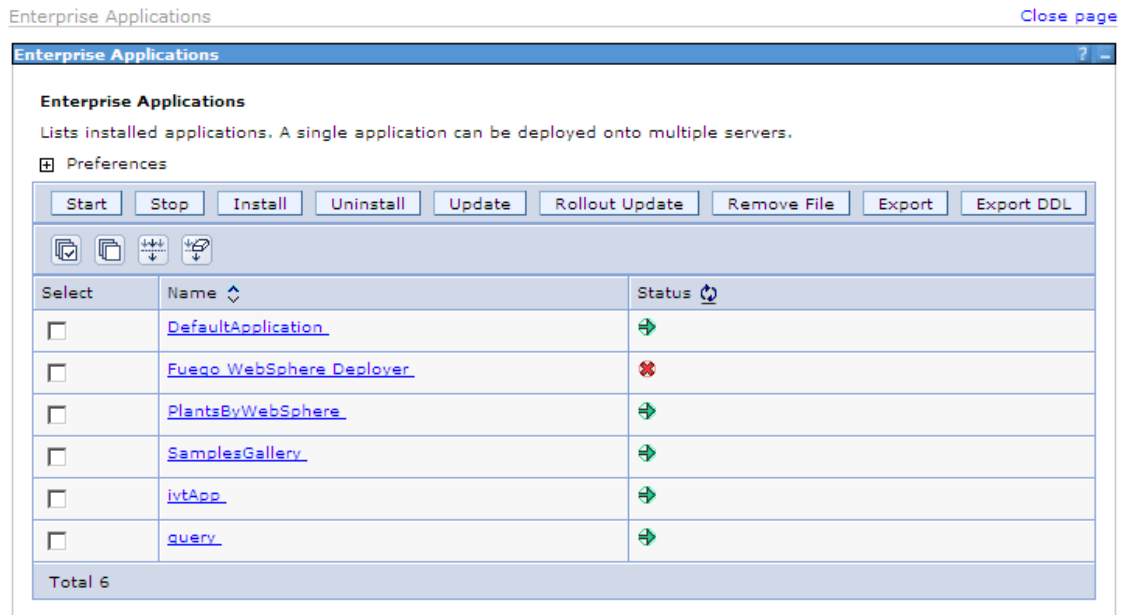
[Save to Master Configuration](#)

To work with installed applications, click the "Manage Applications" button.

[Manage Applications](#)

WebSphere Administrator should now *Save* the application deployment to the master configuration.

Next step is to start the FuegoBPM J2EE Deployer by checking the *Applications -> Enterprise Applications* entry and then click on the *Start* button on the toolbar.



After successfully starting the FuegoBPM J2EE Deployer Application, the application entry should have changed the status to the green arrow

Check FuegoBPM Deployer URL

Check that the FuegoBPM Deployer URL is correct and matches the expected assigned port in WebSphere

1. Within FuegoBPM Web Console
 - a. Go to *Servers -> Application Server (tab)*
 - b. Check the defined FuegoBPM Deployer URL: *host* must be the computer where the WebSphere is running and *port* must match the port defined in WebSphere as explained below.
2. Within WebSphere Console
 - a. Go to *Servers -> Application Servers*

- b. Select the server
- c. Within the *Container Settings* section, select *Web Container Settings*
- d. Click the *Web container transport chains* link
- e. The **port** defined in the FuegoBPM Deployer URL must match the assigned one to *WCInboundDefault*.

Notes for installing in cluster configuration

The WebSphere Administrative Console does not support deploying ears in the Server of the Deployment Manager Node, so it must be deployed using other mechanism as the *wsadmin*, WebSphere ant task or command line tools provided by WebSphere. Although WebSphere Administrative Console does not support installing new applications in the server of the Deployment Manager, it does support updating applications in that server with a new version of the ear.

FuegoBPM Server EAR

FuegoBPM Server EAR

The FuegoBPM J2EE Server is a collection of Services common to all FuegoBPM deployed projects. These services come in the form of an EAR file that is created within FuegoBPM Web Console by FuegoBPM Administrator. This EAR file can then in turn be deployed by the Application Server's Administrator in the Application Server (if you choose to do it manually) or you can automate the deploy using FuegoBPM Deployer, which is the procedure described below.

Click on the **Server** node on the left navigational tree of the Web Console. Click on the Server name to proceed.



Once in the FuegoBPM Server *Basic Configuration* Tab, click on the *J2EE Application Server files (EARs, WARs)*. A panel is opened where the FuegoBPM J2EE Applications can be assembled to be given to the Application Server's Administrator for deployment.

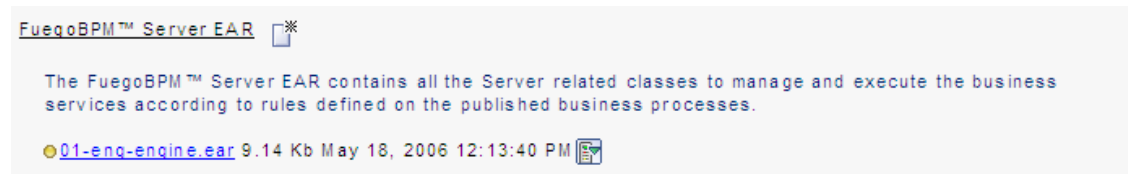
Generate the EAR file

If the Server EAR file has not been generated, you only see the icon to generate the file.




Install the Server

Once you click on the  file generation icon, the link to the file is shown, as uninstalled ( - in yellow), and the icon to install the EAR file in the application server will be provided.



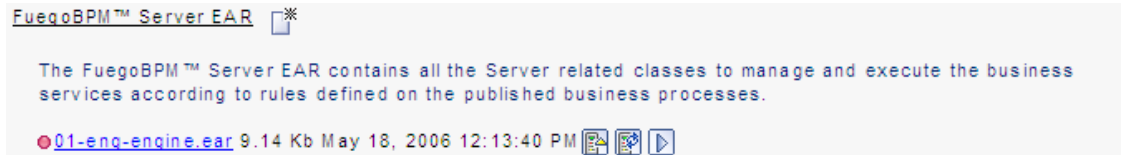
If FuegoBPM Deployer has already been installed, FuegoBPM Administrator can add the Server application directly from FuegoBPM web console.



If the FuegoBPM Server's status does not display, you need to check that the FuegoBPM Deployer is running on the Application Server and the **FuegoBPM Deployer URL** is correctly defined in FuegoBPM Web Console(host and port at *Servers -> Application Server (tab)*).

If you click on the  file installation icon, a message is sent to the FuegoBPM Deployer in order to install the application. This process might take some time. The status will be changed if the installation

succeeds, it is set to **stopped** (● - in red). Besides, a series of possible new operations will be displayed on the side.



Start the Server



Click on the  start icon in order to start the application. If the start succeeds, the status is update to 'running' (● - in green), and the available operations are updated as well. Click on the  stop icon in order to stop the application.



Other Operations

- Click on the  uninstall icon in order to uninstall the application
- Click on the  reinstall icon in order to uninstall and install again the application

FuegoBPM Work Portal Web Application EAR and WAR files

FuegoBPM provides both an EAR file and a WAR file for the FuegoBPM Work Portal Application. The administrator will choose which of them to install in the Application Server.

In J2EE a web application can be package as a WAR (Web Archive)

as well as an EAR (Enterprise ARchive) file. FuegoBPM provides Work Portal and Portal Console in both WAR and EAR formats and only one of them needs to be installed. The preferred one is the EAR file, specially because the FuegoBPM J2EE Deployer is capable of managing it.

Note



having both WAR and EAR files installed may cause conflicts.

To obtain either of these files, open FuegoBPM Web Console and click on **J2EE application server files (EARs, WARs)**, placed inside the server properties link.

If the **EAR file** is the one intended to install in the Application Server, the FuegoBPM Administrator should generate it from the FuegoBPM Web Console Panel and deploy it in the Application server following the indications under the Deploying FuegoBPM Project section. If this file is the one deployed in the application server, it can be managed through the **FuegoBPM Web Console via FuegoBPM Deployer**.

If the **WAR file** is the one intended to install in the Application Server, the FuegoBPM Administrator should first generate the Portal WAR file from the FuegoBPM Web Console Panel. Once the WAR file has been generated, FuegoBPM Administrator should download the WAR and give it to the Application Server Administrator for its deployment using the Application Server Console. The deployment of this FuegoBPM application should follow WAR deployment taking all the defaults.

Provide the generated files to the Application Server Administrator for deployment.

Connecting to the Work Portal

After the EAR or WAR file has been successfully deployed in the Application Server and the Work Portal Application is running, the

next step is to connect to the Work Portal. It is recommended that the project with its processes is first deployed so you can see them all when connecting to the Portal.

You can connect to the Portal with a URL similar to this one:

http://host:port/portal

- **host** : is the host name where the Application Server is deployed (at least one of the nodes on a clustered environment)
- **port**: is the port where the Application Server is listening for incoming HTTP connection.
- **portal**: is the out of the box Web Application name but if you have decided to change its name, you will need to adapt the URL mentioned above.

Installing FuegoBPM Work Portal Web Application

Configuration on WebSphere Server in UX like systems

To assure that Work Portal works properly, the property *java.awt.headless* has to be set to true. If it is not set, for example, the Audit Trail image does not display.

To set this property, from the WebSphere console go to the Application Servers link on the left tree. Edit the application server, and go to **Process Definition->Java Virtual Machine**. Set the Generic JVM arguments property with **-Djava.awt.headless=true**.

Advanced Java virtual machine settings.

Configuration

General Properties

Classpath

Boot Classpath

☐ Verbose JNI

Initial Heap Size

Maximum Heap Size

☐ Run HProf

HProf Arguments

☐ Debug Mode

Debug arguments

Generic JVM arguments

Additional Properties

- [Custom Properties](#)

Help

Field help
Additional command line arguments for the JVM.

Page help
[More information about this page](#)

Installing FuegoBPM Portal Console Web Application

FuegoBPM Administrator should follow the same procedure as for the Work Portal for providing the WAR or EAR file for the Portal Console Web Application and its installation.

Connecting to the Portal Console

After the EAR or WAR file has been successfully deployed in the

Application Server and the Portal Console Application is running, the next step is to connect to the Portal Console.

You can connect to the Portal Console with a URL similar to this one:

http://host:port/portaladmin

- **host** : is the host name where the Application Server is deployed (at least one of the nodes on a clustered environment)
- **port**: is the port where the Application Server is listening for incoming HTTP connection.
- **portaladmin**: is the out of the box Web Application name but if you have decided to change its name, you will need to adapt the URL mentioned above.

Publishing and Deploying Projects

Publish and Deploy activates the process design into a real-time situation where the activities and roles can be fulfilled automatically or by human users.

When you are satisfied with the project design meeting your business's requirements, you can proceed to **Publish & Deploy** it.

Publishing a project

Allows to publish FuegoBPM projects in a directory service. Projects need role values mapping, external variables mapping and other mappings

All processes are grouped in a project when stored to the FDI managed repository and the project has a version number.

Deploying a project

During deployment:

- The processes are associated to a certain Organizational Unit
- The Server is notified that a new process (or a new version of an already deployed process) is available to users associated to that Organizational Unit, so that they can begin working with it. If the end users are currently working with an older copy of the business process in the Work Portal, they will not lose any of the instances in their queues.

Publishing a Project

Before publishing the project from the FuegoBPM Web Console:

1. Export the project from FuegoBPM Studio. Select the *Include versionable libraries only* option.
2. Copy the non-versionable libraries (like database drivers) to:
 - a. **...\webapps\webconsole\WEB-INF\lib**
 - b. **...\ext**
 - c. **...\webapps\portal\WEB-INF\lib** for the built in portal
 - d. If the portal is deployed in a third-party application server, re-create the ear or war file from the FuegoBPM Administration Center and re-deploy it. Alternatively, the non-versionable jars can be copied directly into the **WEB-INF\lib** folder under the portal's deployment folder. Just remember to copy them over again if you ever re-deploy the original *ear o war* file.
3. Publish and deploy the project from the FuegoBPM Web Console

Learn more about Versionables and Non-Versionables libraries in the Administrating Java Class Libraries - JCLs section in the System Administrator Guide.

To publish a project

1. Select the **Published Project** option. A new panel appears with a list of all published projects.
2. Click the **Publish** button. The publish panel appears.

Published projects > Publication source

Publication source

☐ Remote Project

☒ Exported Project
F:\fuegoProjects\DocEx

☐ Vcs
Provider

Publication Properties

☒ Smart publish
When this option is selected, all data defined in the project will be imported to the directory (including organizational units, roles, calendar rules, holiday rules, configurations, variables and views). The needed mappings for roles, configurations and variables will also be resolved automatically.

☐ Increment project version
The version of the project will be incremented to a major version number.

☐ Republish a previous version
You will be able to select the version for which you want to create a new revision. The project must be compatible with the last published revision of the selected version.

☐ Keep generated files
If you select this option, the java files generated during publication won't be deleted from the build directory of the project. This option is useful when debugging.

Deploy Properties

☒ Deploy processes after having published them
You will be able to configure where to deploy each process of this project, and then deploy them.

☐ Import Project's custom Views and Presentations after Deploy
Custom views and custom presentations will be imported to the directory service using the role and variable mappings defined in the next step. This operation only takes place when previous Deploy option is selected.

3. Specify in **Remote project** , the project with its complete path to the directory in which it resides in the server where the Web

Console is installed; or Specify in the **Exported project file** a path to the exported file of the project. The file should reside in any directory visible from the client computer from where the Web Console is being accessed; or Specify the connection data to a VCS repository in which the project has been previously committed in the **VCS** field. If you are publishing a project from ClearCase, you need to specify some extra properties:

[Published Projects](#) > [Publication Source](#) > [ClearCase Properties](#)

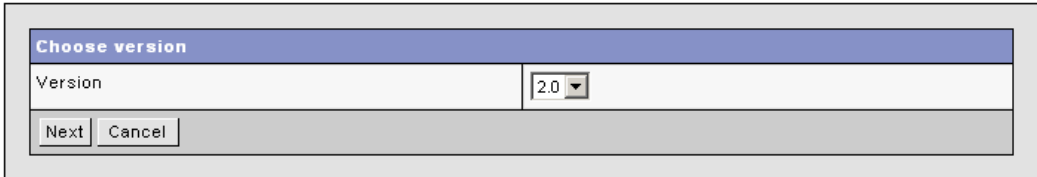


- a. **Path to cleartool program (cleartool.exe)** : cleartool.exe command path.
 - b. **Project Path in Repository**: Directory path in the repository where the view that contains the project to be published has been created. *view_directory/sources/fuego/Project.fpr*
 - c. **Type of View**: Select if it is a snapshot view or a dynamic view.
4. Select the **Smart publish** option if you want to automatically add the organizational info of the project. All the configurations and variables will also be imported to the directory service. It is recommended that you use this option at least for the first time you deploy the project. The published project not only imports all the information defined in the project during development stage, but it also automatically performs all the mappings needed to complete the publication successfully. Once these objects are created in the directory service the **Smart Publish** option will

not update the data if anything is changed in the Studio for these objects.

5. Select the **Increment versions** check box if you want to force an increment of the major version number of all the processes in the project. If not, the publishing task will automatically manage the version numbers.
6. Select the **Republish a previous version** check box if you want to create a new **revision** for any of the published versions of that project. This is mainly applicable if you need to modify a deployed project version that is not the latest one, for example, if you need to introduce a minor change to a deprecated project version (deprecated because a newer version was deployed) so the existing instances can flow according to the new rule you need to introduce. This is valid only if it is a minor change. A new panel appears where you can select the version.

[Published projects](#) > [Publication source](#) > **Choose version**



The project to publish has to be compatible with the latest revision for the selected version. For example if the latest version is 2.0.1, the new revision 2.0.2 is created.

7. Select the **Keep generated files** check box if you want to keep the Java files generated during publication. This is useful when debugging.
8. Select the **Deploy processes after having published them** check box if you want the deploy pane to appear immediately after the publish operation ends.
9. Select the **Import Project's custom Views and Presentations after Deploy** check box if you want custom

views and custom presentations imported to the directory service using the role and variable mappings defined in the next step. This operation only takes place if the previous **Deploy** option is selected.

10. Click **Ok**.

A new panel appears with all processes included in the project.

[Published projects](#) > [Publication source](#) > [Confirm](#) > **Publish process**

Publication Info

Will publish version: 1.0.0

☐ [Processes](#)

Name	Variation	Version
NonParametricRoles	Default	Added
OrderFill	Default	Added
UsingParametricRoles	Default	Added

The files generated during publication will be stored in host 'marvin' under the directory: /home/wc/mt/local/dist/enterprise/tomcat/temp/OrderFillMarine.fpr/build/output--17235121291596329412

☐ [Remarks](#)

☐ [Role mapping](#) ▶

☐ [Variable mapping](#) ▶

☐ [Configuration mapping](#) ▶

1. If necessary, type a comment in the **Remarks** area.
2. Perform the required mappings as necessary: **Roles** defined at design time in the project, and roles that exist in the organizational settings. **Variables** defined as external in the project, and variables defined from the Web Console. **Configurations** defined as external resources to the project, and configurations created in the Web Console. **Business Parameters** defined at design time in the project, and business

parameters created in the Web Console.

[Published projects](#) > [Publication source](#) > [Publish process](#)

Publication Info
Will publish version: 1.0.0
[Processes](#)

Name	Variation	Version
NonParametricRoles	Default	Added
OrderFill	Default	Added
UsingParametricRoles	Default	Added

The files generated during publication will be stored in host 'marvin' under the directory: /home/wc/mt/local/dist/enterprise/tomcat/temp/OrderFillMarine.fpr/build/output--17235121291596329412

[Remarks](#)
[Role mapping](#)

Abstract role	Real role
Account Manager	Account Manager
Corporation Commercial Manager	Corporation Commercial Manager
Corporation Individual Manager	Corporation Individual Manager
Corporation Manager	Corporation Manager
Customer	Customer
Finance Clerk	Finance Clerk
Shipping Clerk	Shipping Clerk

[Variable mapping](#)

Process variable	Real variable
approved	approved
discount	discount
orderAmount	orderAmount
payment	payment

[Configuration mapping](#)


Process configuration	Real configuration
J2EE	J2EE
OrdersDB	OrdersDB

Note

 If you are publishing a new **revision** of the project, you cannot change the associated role mapping.

3. Before clicking the **Publish** button, ensure that there is a green flag next to the **Role**, **Variable**, Business Parameters and **Configuration** mappings.


Note

 If the mappings cannot be resolved automatically, a red flag indicates that there are mappings that need to be manually defined before project publication. You can create all of the missing elements, or only part of them and map the rest manually. After all the flags appear green, click the **Publish** button.

Once the publication is completed, the project is compiled and the processes are included in the definitions list.

The mappings (roles, external variables, business parameters and configurations) and remarks are stored in the published project.

Warning

 Mappings are applicable and will work only if the **Smart Publish** was enabled and these objects HAVE NOT BEEN created previously.

When a project is published, it is compiled into a Java class, which has all the necessary code to implement the functionality defined during design.

If the **Deploy processes after having published them** option was checked in the first step, the Deploy panel appears. Choose the Server, organizational unit, views and archiving settings.

Published projects > Publication source > Confirm > Publish process > **Deploy**

Deployment topology

Show deployment grouped by: Server

Server: DocServer

Version: 1.0 Revision: 0

Name	Variation	OU	Views Generation	Archiving	Status	Action
NonParametricRoles	Default		Unified Inbox	<input type="checkbox"/> Enable archiving		Deploy
OrderFill	Default		Unified Inbox	<input type="checkbox"/> Enable archiving		Deploy
UsingParametricRoles	Default		Unified Inbox	<input type="checkbox"/> Enable archiving		Deploy

OK Cancel

All the projects that are published are listed by name, with a link to the **history** of each project and another link to view/edit the **deployment** state.

The history shows the list or **versions**, in descendent order. It shows, as well, the **number of processes** and **number of published revisions**.

Published projects

Page: 1/1 - Total: 1				Filter	Preferences
Publish Unpublish					
<input type="checkbox"/>	▲ Name	History		Deployment	
<input type="checkbox"/>	OrderFill	1.0.0		Not deployed	


Click here to see the project's history

[Published projects](#) > **OrderFill**

Page: 1/1 - Total: 1				Filter	Preferences
Unpublish Deploy					
<input type="checkbox"/>	▲ Version	Processes		Revisions	
<input type="checkbox"/>	1.0	3		1	


If you select the **number of processes**, a new list is displayed, showing the publication info by process name.

Published projects > OrderFill

 Page: 1/1 - Total: 1

[Filter](#) [Preferences](#)

[Unpublish](#) [Deploy](#)

	Version	Processes	Revisions
<input type="checkbox"/>	1.0	3	1

Click here to see the project content

Published projects > OrderFill > 1.0.0

Process	Variation	Author	Creation time
NonParametricRoles	Default	carolina	2004-07-30 15:56:29-03
OrderFill	Default	Administrator	2004-07-30 15:56:35-03
UsingParametricRoles	Default	carolina	2004-07-30 15:56:38-03

Abstract role	Real role
Finance Clerk	Finance Clerk
Corporation Individual Manager	Corporation Individual Manager
Shipping Clerk	Shipping Clerk
Corporation Commercial Manager	Corporation Commercial Manager
Customer	Customer
Account Manager	Account Manager
Corporation Manager	Corporation Manager

Process variable	Real variable
approved	approved
discount	discount
orderAmount	orderAmount
payment	payment

Process configuration	Real configuration
J2EE	J2EE
OrdersDB	OrdersDB

If you select the number of **published revisions**, a list of revisions in descendent order is shown, with the number of processes, from where you are able to see the previous view.

Deploying a Project

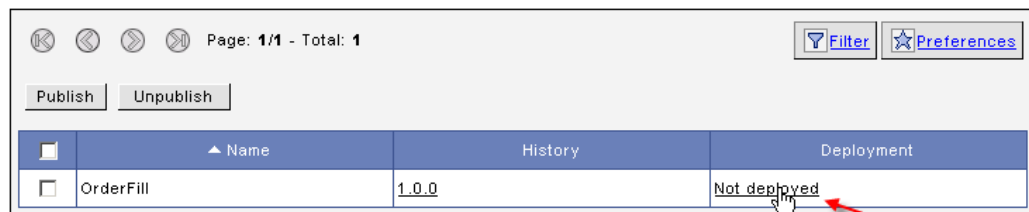
To enable users and FuegoBPM Servers to run processes from a

project, the processes must be deployed in a specific Server. Before this is done, process definitions are available and stored in a repository that can be seen only from the Web Console connected to it. After the deployment, users with permission can see the processes as available applications, and Servers can start running instances on them.

To deploy a project

1. Click on the **Deployment column** for the project you want to deploy.

Published projects



Published projects

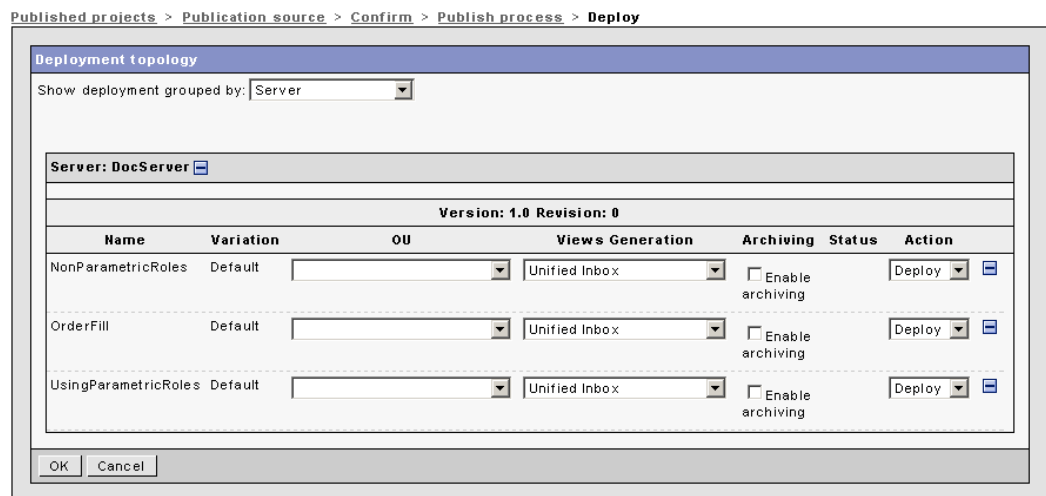
Page: 1/1 - Total: 1

[Filter](#) [Preferences](#)

[Publish](#) [Unpublish](#)

	Name	History	Deployment
<input type="checkbox"/>	OrderFill	1.0.0	Not deployed

2. Click the **Deploy** button. The deploy panel displays.



Published projects > Publication source > Confirm > Publish process > Deploy

Deployment topology

Show deployment grouped by:

Server: DocServer


Version: 1.0 Revision: 0

Name	Variation	OU	Views Generation	Archiving	Status	Action
NonParametricRoles	Default	<input type="text"/>	Unified Inbox	<input type="checkbox"/> Enable archiving	Deploy	<input type="button" value="Deploy"/>
OrderFill	Default	<input type="text"/>	Unified Inbox	<input type="checkbox"/> Enable archiving	Deploy	<input type="button" value="Deploy"/>
UsingParametricRoles	Default	<input type="text"/>	Unified Inbox	<input type="checkbox"/> Enable archiving	Deploy	<input type="button" value="Deploy"/>

3. Choose the organizational unit (to deploy for all organization leave it empty)

4. Choose between basic view configurations: **Unified Inbox** shows all instances to the user in a folder called Inbox. The instance information will allow users to identify in which process it is running. **By process** distributes the instances to their corresponding processes. **By process and activity** includes not only the instances under each process entry but also inside the activities that contains them. If the process have grab activities, the user will only see the grab activity view, if views are created **By process and activity**. Otherwise, the user will only be able to grab instances, from the Search results panel or if you manually create a custom view for the grab activity of type *Activity Instances*.
5. Choose the archiving settings. If you enable archiving for this particular process, you will be able to decide as well wether you want to archive attachments and notes.
6. Click the **OK** button.

Note

 If you have more than one Server, you have to select in which Server you want to deploy your project.

All deployed processes are shown, as well as their status.

Published projects > Deploy

Deployment topology

Show deployment grouped by: Server


Server: DocServer					
Version: 1.0 Revision: 0					
Name	Variation	OU	Views	Generation	Archiving
NonParametricRoles	Default	Organization	Unified Inbox		<input type="checkbox"/> Enable archiving
OrderFill	Default	Organization	Unified Inbox		<input type="checkbox"/> Enable archiving
UsingParametricRoles	Default	Organization	Unified Inbox		<input type="checkbox"/> Enable archiving

It is possible to group the deployment information by Server or by Organizational Unit.

If you are deploying the processes into an EJB based Server , you must generate the EARs files.

See Maintaining a Server for further information.

Note

 If your project has non-versionable Java Class Libraries (JCLs), you will need to copy them to FuegoBPM's installation directory. Please refer to the System Administration Guide - Java Class Libraries for the detailed information.

FuegoBPM Project EAR

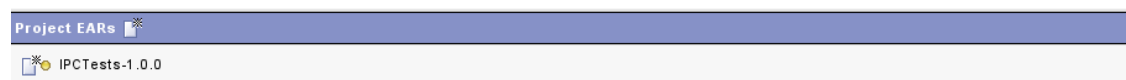
FuegoBPM Project EAR files



After successfully publishing and deploying the FuegoBPM Project into the FuegoBPM Directory Service, it is necessary to deploy the FuegoBPM Project into the Application Server.

Click on the **Server** node on the left navigational tree of the Web Console. Click on the Server name where the FuegoBPM Project has been deployed.

Click on the *J2EE Application Server files (EARs, WARs)*. A panel is opened where the FuegoBPM J2EE Applications can be assembled to be given to the Application Server's Administrator for deployment.

If the Project EAR files have not been generated, you will only see the icon to generate them.





Once you click on the  file generation icon, the link to the file will be shown, as uninstalled (), and the icon to install the EAR file in






the application server will be provided.



If FuegoBPM Deployer has already been installed, FuegoBPM Administrator can add the FuegoBPM project directly from FuegoBPM web console.

If you click on the  file installation icon, a message is sent to the FuegoBPM Deployer in order to install the application. The status will be changed if the installation succeeds, set as 'stopped' () . Besides, a series of possible new operations will be displayed on the side.



- Click on the  uninstall icon in order to uninstall the application
- Click on the  reinstall icon in order to uninstall and install again the application
- Click on the  start icon in order to start the application. If the start succeeds, the status is update to 'running' () , and the available operations are updated as well. Click on the  stop icon in order to stop the application.



Chapter 8. Additional steps

Additional steps

See *Appendix G: Known issues* to review if any further step is required.

Chapter 9. Logging and Tracing

Logging and Tracing

If any problem comes up when starting the engine you can check the WebSphere log file, as well as the Engine log file.

1. Within WebSphere Console
 - a. Go to *Servers -> Application Servers*
 - b. Select the server
 - c. Within the *Troubleshooting* section, select *Logging and Tracing*
 - d. Click the *JVM Logs* link
 - e. From the *Runtime* tab, check the **SystemOut.log** and **SystemErr.log** files.
2. Within FuegoBPM Web Console
 - a. Check the Engine log if already created.

The WebSphere log file can be read from
`$INSTALL_WEBSPHERE_DIR\AppServer\profiles\default\logs\servername`

Chapter 10. Troubleshooting

Troubleshooting

Error Message or exception:

```
XXXX Project fails to deploy and gives the following errors.  
  
[5/11/06 17:07:50:746 EDT] 00000037 WorkspaceMast  
E WKSP0022E:  
File validation failed --java.io.IOException:  
The URI length is greater than the Windows  
limit of 259 characters.  
Unable to create or pdate file:  
D:\\WebSphere\\AppServer\\profiles\\default  
\\wstemp\\Script10b254613ca  
\\workspace\\cells\\PUSEHWUNode01Cell\\applications  
\\02-proj-FuegoWASServer-XXXX-1.0.ear\\deployments  
\\02-proj-FuegoWASServr-XXXX-1.0\\project-XXXX-1.0.jar  
\\META-INF\\ibm_ejbext.properties
```

Resolution: URI length exceeds the Windows limit of 259 characters. Configure a new workspace directory with a short path.

1. Uninstall all applications already installed in websphere through the FuegoBPM Web Console.
2. Create a folder for example C:/temp.
3. Go to the Websphere work console and click *Servers -> Application Servers -> server name -> Java and Process Management -> Process Definition -> Java Virtual Machine*. Then in **Generic JVM arguments** enter for example: `-Dworkspace.user.root=C:/temp`
4. Restart the Application Server

[Application servers](#) > [server1](#) > [Process Definition](#) > Java Virtual Machine

Advanced Java virtual machine settings.

Configuration

General Properties	Additional Properties
<p>Classpath</p> <input type="text"/>	<p>■ Custom Properties</p>
<p>Boot Classpath</p> <input type="text"/>	
<p><input type="checkbox"/> Verbose class loading</p>	

...

Maximum Heap Size	<input type="text"/>
<input type="checkbox"/> Run HProf	
HProf Arguments	<input type="text"/>
<input type="checkbox"/> Debug Mode	
Debug arguments	<input type="text" value="-Djava.compiler=NONE -Xdeb"/>
Generic JVM arguments	<input type="text" value="-Dworkspace.user.root=C:\W"/>
Executable JAR file name	<input type="text"/>
<input type="checkbox"/> Disable JIT	
<input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>	

Error Message or exception:

```
... 25 more
Caused by: fuego.papi.impl.EngineNotAvailableException:
    Cannot access Server 'engine' EJB at 'engines/engine'.
Reason: Unable to load class:
    fuego.papi.impl.j2ee._EJBSecureEngineHome_Stub
Please check the EJB stubs are available to the application
and that there are no classloader conflicts.
    at fuego.papi.impl.EngineNotAvailableException.
```

```
classCastOnNarrow(EngineNotAvailableException.java:99)
at fuego.papi.impl.j2ee.J2EEEngineAccessImpl.
  getEngineHome(J2EEEngineAccessImpl.java:288)
at fuego.papi.impl.j2ee.J2EEEngineAccessImpl.
  getSecureEngineInterface(J2EEEngineAccessImpl.java:353)
at fuego.papi.impl.j2ee.J2EEEngineAccessImpl.
  createSecureEngine(J2EEEngineAccessImpl.java:177)
at fuego.papi.impl.ProcessServiceImpl.createSecureEngine
  (ProcessServiceImpl.java:1268)
at fuego.papi.impl.ProcessServiceNESessionImpl.
  getSecureEngine(ProcessServiceNESessionImpl.java:2091)
at fuego.papi.impl.ProcessServiceImpl.createProcess
  (ProcessServiceImpl.java:1574)
... 31 more
Caused by: java.lang.ClassCastException: Unable to load
class:
  fuego.papi.impl.j2ee._EJBSecureEngineHome_Stub
at com.ibm.rmi.javax.rmi.PortableRemoteObject.narrow
  (PortableRemoteObject.java:366)
at javax.rmi.PortableRemoteObject.narrow
  (PortableRemoteObject.java:157)
at fuego.papi.impl.j2ee.J2EEEngineAccessImpl.
  getEngineHome(J2EEEngineAccessImpl.java:283)
... 36 more
```

Resolution:

The FuegoBPM for J2EE uses EJBs as part of its implementation. In J2EE the client of EJBs needs a Stub class for the beans. Other application servers than WebSphere 6 make these stubs automatically available at runtime to all the clients (in our case, the client is FuegoBPM Work Portal). In WebSphere 6 this does not happen automatically and FuegoBPM provides this jar at *\$FUEGO\$/j2ee/j2ee/websphere/was6stubs/fuegoj2ee-was6-stubs.jar*. Make sure that this jar has been installed correctly in the shared libraries as describe in section *Installing FuegoBPM Libraries for WebSphere*.

Error Message or exception:

```
java.sql.SQLException: Attempt to set Escape Processing
after SQL is already processed.
```

Resolution: In most of the cases, this exception shows up because you are using a JDBC Driver does is not the same one bundled with the Oracle Server Installation. You may use ojdbc14.jar from the Oracle Server's installation directory into WebSphere's environment.

Error Message or Exception:

```
[2/11/05 14:40:57:232 CST] 7c63b6b0 XARminst
E WTRN0037W: The transaction service encountered
an error on an xa_recover operation.
The resource was J2CXAResourceInfo :
cfName = Fuego Server Data Source
configProps = [Deployed Resource Adapter Properties]
OptionC_authDataAlias java.lang.String
                gandalf/Fuego55WSEng
UserName      java.lang.String Fuego55WSEng
Password      java.lang.String *****
TransactionResourceRegistration java.lang.String
                dynamic
InactiveConnectionSupport      java.lang.Boolean true
secureMode boolean true

...
recoveryClasspath =
f:\Tools\WebSphere\AppServer/lib/rsadapter.rar
. The error code was XAER_RMERR.
The exception stack trace follows:
    javax.transaction.xa.XAException
at oracle.jdbc.xa.OracleXAResource.recover
    (OracleXAResource.java:626)
at com.ibm.ws.rsadapter.spi.WSRdbXAResourceImpl.
    recover (WSRdbXAResourceImpl.java:672)
at com.ibm.ws.Transaction.JTA.XARminst.recover
    (XARminst.java:130)
at com.ibm.ws.Transaction.JTA.XARecoveryData.recover
    (XARecoveryData.java(Compiled Code))
at com.ibm.ws.Transaction.JTA.RecoveryManager.resync
    (RecoveryManager.java:1415)
at com.ibm.ws.Transaction.JTA.ResyncThread.run
    (RecoveryManager.java:1448)
```

Resolution: WebSphere's Administrator should ask Oracle's Administrator to run the following statement so that Oracle can manage recoverability from uncompleted or rolled back transactions. This SQL statement can be run from SQL-Plus as SYSOPER or SYSDBA:

```
grant select on DBA_PENDING_TRANSACTIONS to FUEGO55WASDB;  
grant select on DBA_PENDING_TRANSACTIONS to FUEGO55WASFDI;
```

Error Message or Exception:

```
WSRdbDataSour I DSRA8205I: JDBC driver name :  
                        Oracle JDBC driver  
WSRdbDataSour I DSRA8206I: JDBC driver version:  
                        9.2.0.1.0  
WSRdbXaResour E DSRA0304E: XAException occurred.  
                        XAException contents and details are:  
The XA Error is          : -3  
The XA Error message is  : A resource manager error has  
                        occurred in the transaction branch.  
The Oracle Error code is : 65535  
The Oracle Error message is: Internal XA Error  
The cause is            : null.  
WSRdbXaResour E DSRA0302E: XAException occurred.  
                        Error code is: XAER_RMERR.  
Exception is: <null>  
RegisteredRes E WTRN0078E: An attempt by the transaction  
                        manager to call start  
                        on a transactional resource has resulted in an error.  
                        The error code was XAER_RMERR.  
                        The exception stack trace follows:  
oracle.jdbc.xa.OracleXAException  
at oracle.jdbc.xa.OracleXAResource.checkError  
                        (OracleXAResource.java:1157)  
at oracle.jdbc.xa.client.OracleXAResource.start  
                        (OracleXAResource.java:295)  
at com.ibm.ws.rsadapter.spi.WSRdbXaResourceImpl.start  
                        (WSRdbXaResourceImpl.java:927)  
at com.ibm.ejs.j2c.XATransactionWrapper.start  
                        (XATransactionWrapper.java:1267)  
at com.ibm.ws.Transaction.JTA.JTAResourceBase.start  
                        (JTAResourceBase.java:164)  
at com.ibm.ws.Transaction.JTA.RegisteredResources.  
                        startRes(RegisteredResources.java:389)  
at com.ibm.ws.Transaction.JTA.TransactionImpl.  
                        enlistResource(TransactionImpl.java:1903)  
at com.ibm.ws.Transaction.JTA.TranManagerSet.enlist  
                        (TranManagerSet.java:494)
```

Resolution: The error is that the transaction manager was unable

to convert the transaction to an XA transaction. Specifically, Oracle returned an error XAER_RMERR, error code number 65535. ORA-65535 is not a valid error code and XAER_RMERR essentially means that something went wrong with XA. What this indicates is that your Oracle database is not configured to support XA transactions. Therefore, when the WebSphere Application Server transaction manager instructs the Oracle transaction manager to participate in this XA transaction, Oracle cannot comply and throws this exception.

To check if the Oracle RDBMS used as the backend transactional store for FuegoBPM Server, we can query Oracle's Synonyms table and check if the JAVA_XA entry is located. The following query can be executed:

```
select * from all_synonyms where synonym_name
                                like '%JAVA_XA%';
```

The solution is to run two scripts that are included in the Oracle install. This will likely need to be performed by your Oracle DBA, since you must be logged into Oracle as SYSOPER or SYSDBA in order to have the necessary permissions for these scripts to work. The scripts are:

- directory: \$ORACLE_HOME\javavm\install
 - file: initxa.sql
 - file: initjvm.sql

The *initxa.sql* script configures the database for XA. Once it runs successfully, your database is configured for XA. The script may run successfully the first time you try. Unfortunately, it probably will not run successfully because some of the database's memory spaces are

too small. To fix this, run the *initjvm.sql* script. It will probably fail too, but in doing so it will indicate which parameters need to be adjusted. The parameters are stored in this file:

- directory: \$ORACLE_HOME\\database
- file: initDATABASE_SID.ora

This table shows two parameters that typically need to be increased. Your particular database configuration may require adjusting the following parameters: `java_pool_size`, `shared_pool_size`.

Error Message or Exception:

```
FreePool      E J2CA0046E: Method
               createManagedConnctionWithMCWrapper caught
               an exception during
creation of the ManagedConnection for resource
               JMS$Fuego Topic
Factory$JMSManagedConnection@981672523, throwing
               ResourceAllocationException.  Original exception:
javax.resource.spi.ResourceAdapterInternalException:
               Failed to create session
               at com.ibm.ejs.jms.JMSCMUtils.mapToResourceException
               (JMSCMUtils.java:125)
...
Next Linked Exception:
com.ibm.mq.jms.JMSNotSupportedException: MQJMS6311:
               Transport type 'DIRECT' within a transaction is not
supported.
               at com.ibm.mq.jms.SessionImpl.(SessionImpl.java:250)
               ...
               at com.ibm.ejs.jms.JMSTopicConnectionHandle.
                               createTopicSession
               (JMSTopicConnectionHandle.java:249)
               at fuego.ejbengine.service.EJBNotifyQueue.sendNews
               (EJBNotifyQueue.java:157)
               at fuego.ejbengine.service.EJBNotifyQueue.sendNewsToAll
               (EJBNotifyQueue.java:67)
               at fuego.ejbengine.service.EJBNotifyQueue.updatePapiFrom
               Directory(EJBNotifyQueue.java:104)
```

Resolution: The WebSphere Topic Connection Factory may not be properly configured. Make sure the Port property in the Topic Connection Factory has been set to *QUEUED*.

Error Message or Exception:

```
[7/29/04 16:15:52:741 ART] 15127a46 ExceptionUtil
E CNTR0020E: Non-application exception occurred
while processing method "sendLaterItems" on bean
"BeanId(01-fuego ear#engine-engine1.jar
#engine-startup-engine1, null)".
Exception data:
fuego.server.exception.EngineRuntimeException:
Engine runtime exception. Engine runtime exception.
MQJMS2008: failed to open MQ queue
at fuego.ejbengine.EJBToDoQueue.sendLaterItems
(EJBToDoQueue.java:184)
at fuego.ejbengine.ejb.EngineStartupBean.
sendLaterItems(EngineStartupBean.java:47)
at fuego.ejbengine.interfaces.
EJSLocalStatelessengine_startup_engine1_4aea5d25.
sendLaterItems(Unknown Source)
at fuego.ejbengine.servlet.
SchedulerServlet$SendItemsTask.run
(SchedulerServlet.java:130)
at java.util.TimerThread.mainLoop
(Timer.java(Compiled Code))
at java.util.TimerThread.run(Timer.java:396)
Caused by: fuego.server.exception.EngineRuntimeException:
Engine runtime exception. MQJMS2008:
failed to open MQ queue
at fuego.ejbengine.EJBToDoQueue.sendItem
(EJBToDoQueue.java:272)
at fuego.ejbengine.EJBToDoQueue.access$000
(EJBToDoQueue.java:39)
at fuego.ejbengine.EJBToDoQueue$2.execute
(EJBToDoQueue.java:166)
at fuego.server.execution.
EngineExecution$AtomicExecutionTA.
runTransaction(EngineExecution.java:330)
at fuego.transaction.TransactionAction.
startBaseTransaction
(TransactionAction.java:448)
at fuego.transaction.TransactionAction.startTransaction
(TransactionAction.java:515)
at fuego.transaction.TransactionAction.start
(TransactionAction.java:176)
at fuego.server.execution.EngineExecution.
executeImmediate
```



```

        (EngineExecution.java:137)
    at fuego.server.execution.EngineExecution.
        executeAutomaticWork(EngineExecution.java:60)
    at fuego.ejbengine.EJBToDoQueue.
        sendLaterItems(EJBToDoQueue.java:181)
    ... 5 more
Caused by: javax.jms.InvalidDestinationException:
    MQJMS2008: failed to open MQ queue
    at com.ibm.mq.jms.MQQueueSession.
        getQueueOpenException(MQQueueSession.java:935)
    at com.ibm.mq.jms.MQQueueSession.
        getOutputQueue(MQQueueSession.java:876)
    at com.ibm.mq.jms.MQQueueSession.
        createSender(MQQueueSession.java:205)
    at com.ibm.ejs.jms.JMSQueueSenderHandle.
        (JMSQueueSenderHandle.java:86)
    at com.ibm.ejs.jms.JMSQueueSessionHandle.
        createSender(JMSQueueSessionHandle.java:229)
    at fuego.ejbengine.EJBToDoQueue.sendItem
        (EJBToDoQueue.java:261)
    ... 14 more

```

Resolution: WebSphere's Administrator may have not to added the Queue name (not the jndi name) to the jmsserver queue names list (in *Application Servers/[server_name]/Server Components/JMS Server/Queue names*).

Error Message or Exception:

```

[12/11/04 15:52:40:457 CST] 7cfb3066 WSJdbcConnect
    W DSRA8650W: Error closing a JDBC child wrapper,
    com.ibm.ws.rsadapter.jdbc.
    WSJdbcPreparedStatement@6c40706a
java.sql.SQLException: Attempt to set Escape Processing
    after SQL is already processed
    at oracle.jdbc.driver.DatabaseError.throwSQLException
        (DatabaseError.java:125)
    at oracle.jdbc.driver.DatabaseError.throwSQLException
        (DatabaseError.java:162)
    at oracle.jdbc.driver.DatabaseError.throwSQLException
        (DatabaseError.java:227)
    at oracle.jdbc.driver.OracleSql.setEscapeProcessing
        (OracleSql.java:343)
    at oracle.jdbc.driver.OraclePreparedStatement.
        setEscapeProcessing
        (OraclePreparedStatement.java:3190)
    at com.ibm.websphere.rsadapter.OracleDataStoreHelper.

```

```
doStatementCleanup(OracleDataStoreHelper.java:287)
at com.ibm.ws.rsadapter.jdbc.WSJdbcPreparedStatement.
  closeWrapper(WSJdbcPreparedStatement.java:298)
at com.ibm.ws.rsadapter.jdbc.WSJdbcObject.close
  (WSJdbcObject.java:126)
...
```

Resolution: Check that the oracle driver version matches the version of the oracle database.

Error Message or Exception:

```
Get a Signal 11 in the rmic during deploy.
```

Resolution: Check that the server where you are deploying has the DISPLAY variable set.

Error Message or Exception:

```
XXX_JDBC_DRIVER_PATH during new data source config.
in a cluster.
```

Resolution: Check XXX_JDBC_DRIVER_PATH variable on all nodes and at the cell level, by default it is left empty at all nodes, and if you override it at cell level, the variable is shadowed by the empty definition at node level.

Error Message or Exception:

```
java.sql.SQLException: Attempt to set Escape Processing
after SQL is already processed.
```

Resolution: Check that the oracle driver version matches the version of the oracle database.

Error Message or Exception:

```
SRVE0020E: Servlet Error startup: Failed to load servlet:  
java.lang.ClassCastException: java.sql.Timestamp
```

Resolution: Surely a driver version problem. Try downloading last version of FuegoBPM jdbc driver.

Error Message or Exception:

```
Error ("java.lang.RuntimeException: The service URL remote:  
//xxxx/schema=xxxx,sid=xxxx differs from passport URL  
remote://yyyy/schema=yyyy,sid=yyyy");
```

Resolution: This error may arise to users while logging in to portal. The most probable reason is that FuegoBPM Directory URL was changed and there's still one application that has a reference to the old one. If FDI schema or provider is changed, all applications should be deployed again, and configuration data to FDI repository updated in WebSphere console if necessary (for instance, data source settings).

Chapter 11. Appendix A: Creating FuegoBPM Directory Service Database using SQL-Plus

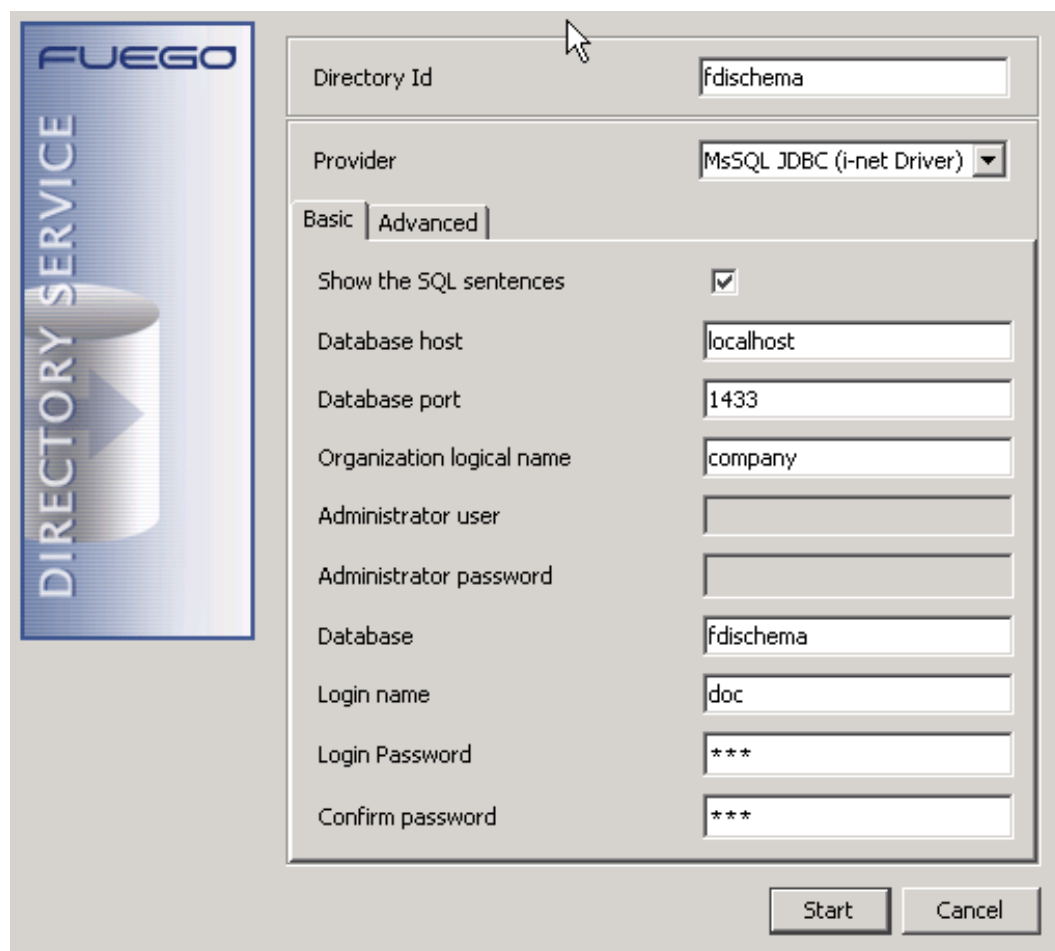
Getting the schema SQL creation script

If it is required to generate the FDI schema on the database from outside the FuegoBPM Enterprise Administration Center, you can get the creation SQL script.

When the Wizard is started and if you selected to *Show the SQL sentences* when creating the Directory Service, a new panel containing the SQL Script to be executed by Oracle's Administrator is displayed.

To get the SQL script

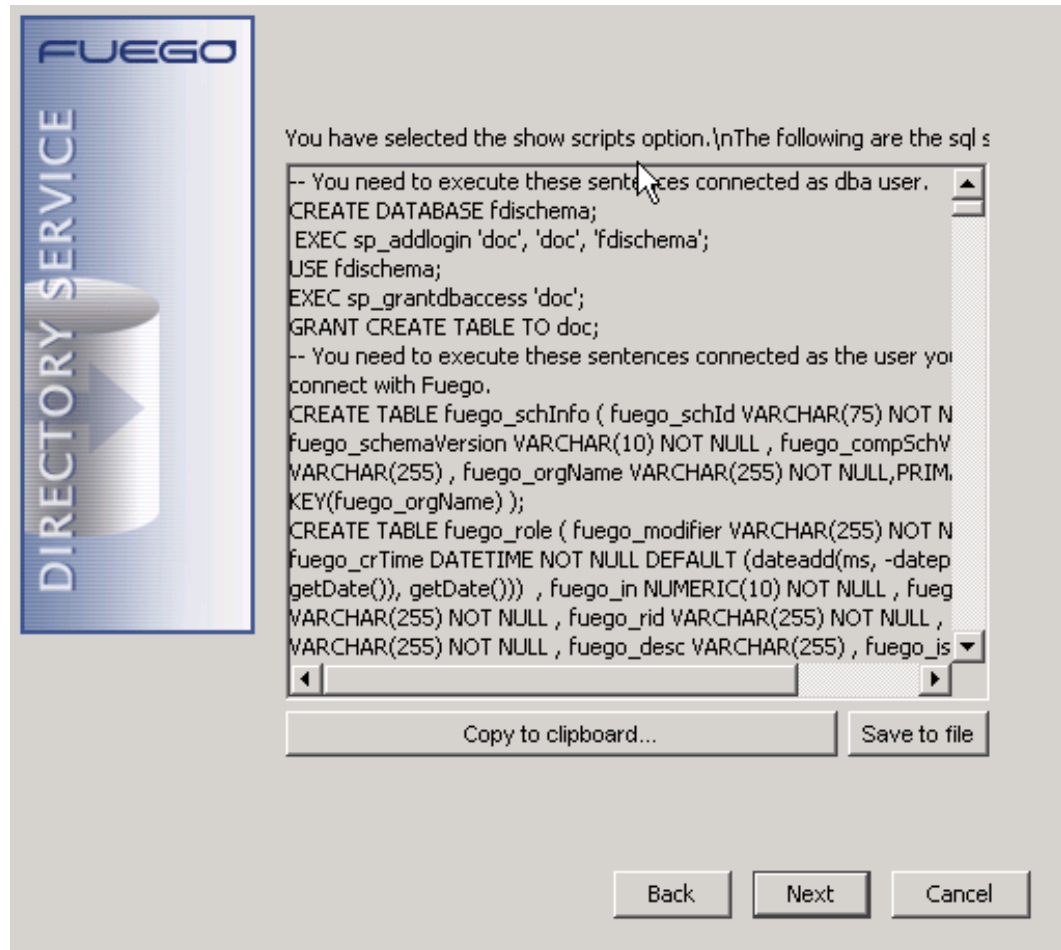
1. Follow the steps as if you were creating the FDI schema from the Administration Center.
2. After filling the fields according to the database, select the *Show SQL script* check box. The *Administrator User and password* are disabled. Click the **Start** button.



The image shows a configuration window for the Fuego Directory Service. On the left is a vertical banner with the 'FUEGO' logo at the top and 'DIRECTORY SERVICE' written vertically below it. The main window has a title bar and contains the following fields and controls:

- Directory Id:** A text field containing 'fdiscema'.
- Provider:** A dropdown menu showing 'MsSQL JDBC (i-net Driver)'.
- Tabs:** Two tabs, 'Basic' (selected) and 'Advanced'.
- Show the SQL sentences:** A checkbox that is checked.
- Database host:** A text field containing 'localhost'.
- Database port:** A text field containing '1433'.
- Organization logical name:** A text field containing 'company'.
- Administrator user:** An empty text field.
- Administrator password:** An empty text field.
- Database:** A text field containing 'fdiscema'.
- Login name:** A text field containing 'doc'.
- Login Password:** A text field containing '***'.
- Confirm password:** A text field containing '***'.
- Buttons:** 'Start' and 'Cancel' buttons at the bottom right.

3. A screen with the sql scripts is displayed.



4. You can *copy to the clipboard* or *save to a file* the script. This script only creates the database schema. Give the scripts to the database administrator, so he creates it, before going on the next step. The script has to be executed by a user with database creation permission. At this moment, you should wait until the script is properly and successfully executed by the Database Administrator. After the script has been run, you can continue with the wizard creation to initialize the FuegoBPM Directory Service Databases.
5. After the database is created, click **Next** in this screen. The following screen explains that before going on with the initialization the schema has to be created. Click **Next** to proceed with the initialization.

Running the schema SQL creation script in Oracle SQL-Plus

After saving the SQL script to create the FuegoBPM Directory Service structure manually, you need the Oracle database Administrator to run this script.

For consistency, the SQL script is consolidated into one single file. However, separate its content into 2 SQL files that will be executed through SQL Plus with different Oracle Users.

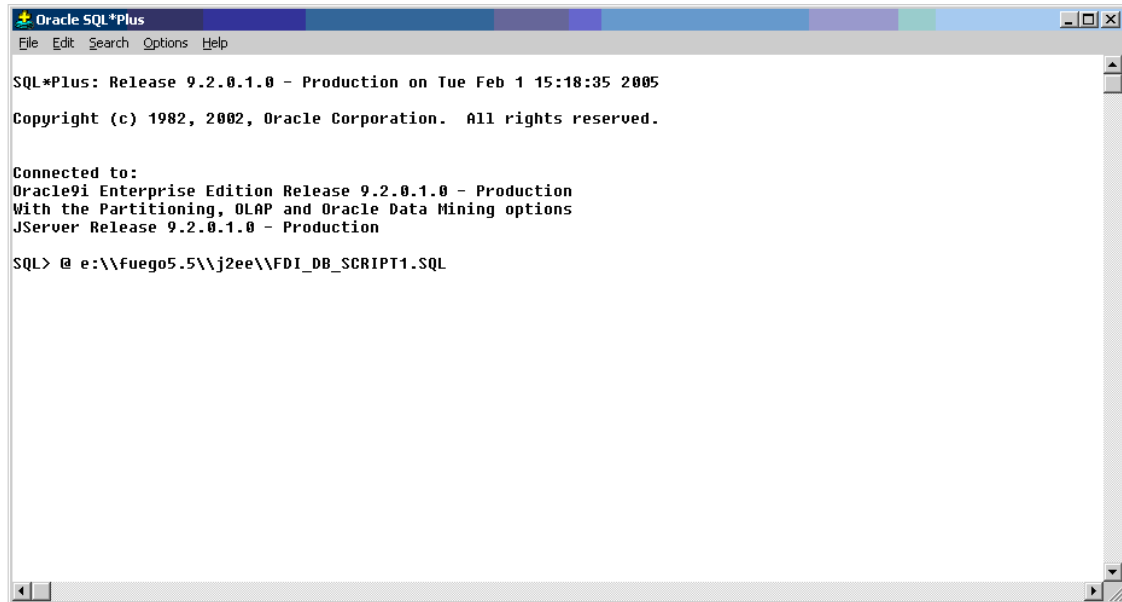
The first SQL script (the one that will be executed by an Oracle User with permission to create Oracle User and tables) should contain the first half of the original SQL script. This is properly delimited and documented in the SQL Script. You can copy the content of this file into a SQL Script file called *FDI_DB_SCRIPT1.sql*.

The second part of the SQL Script should be copied into a new file called *FDI_DB_SCRIPT2.sql*. This second script should be executed with the Oracle User created by *FDI_DB_SCRIPT1* SQL Script. For our example, assume these 2 files were created in FuegoBPM installation directory.

To begin the procedure, the Oracle's Administrator should connect to Oracle using SQL-Plus Editor with a user that has enough permission to create Oracle Users as well as creating tables.

After successfully logging in, Oracle's Administrator should run *FDI_DB_SCRIPT1.sql* file that will create the FuegoBPM Directory Service Oracle User and its tables.

This SQL Script can be executed from SQL Plus as follows:

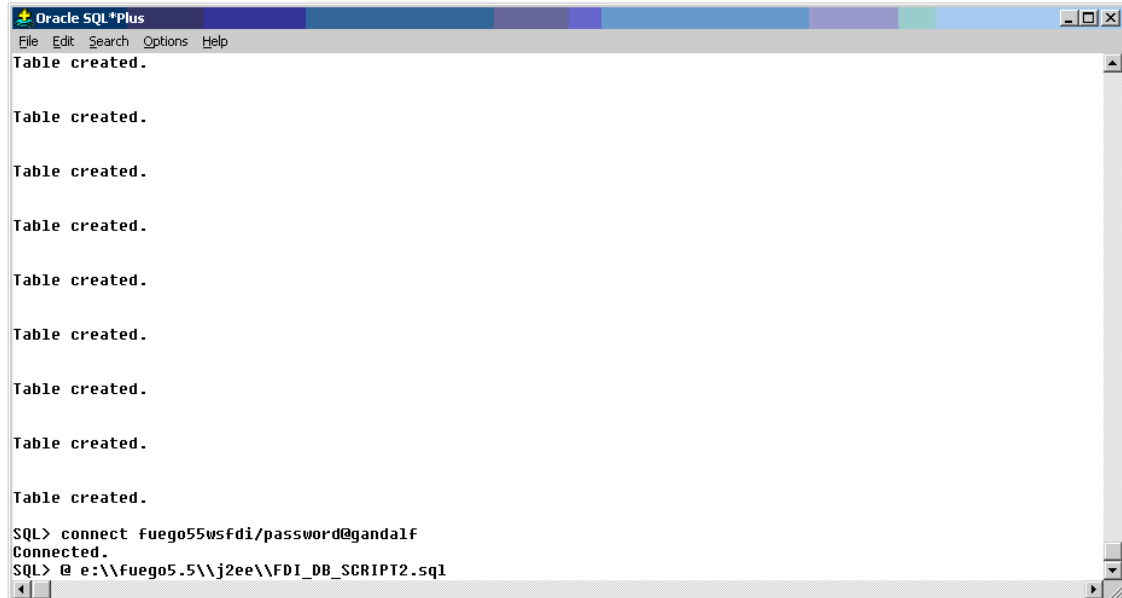


The result of executing this script should be a series of messages with *Table created*. Oracle's Administrator should make sure there are no errors while executing this first SQL script since the second script relies on a successful creation of the FuegoBPM Directory Service Oracle User and its tables.

The next step is to connect with the recently created Oracle User to execute the second script. This script will create needed Oracle Sequences and triggers that enforce FuegoBPM Directory Service Data integrity and consistency. The screenshot below depicts this panel.

```
SQL> connect fuego55wsfdi/password@gandalf
```

After successfully connecting, Oracle's Administrator should execute the second script in the same way the first one was executed. This is shown in the figure below.

A screenshot of the Oracle SQL*Plus command-line interface. The window has a title bar 'Oracle SQL*Plus' and a menu bar with 'File', 'Edit', 'Search', 'Options', and 'Help'. The main text area displays a series of 'Table created.' messages, indicating the successful creation of multiple tables. At the bottom, the user has entered the command 'SQL> connect fuego55wsfdi/password@gandalf', resulting in 'Connected.' and a new prompt 'SQL> @ e:\\fuego5.5\\j2ee\\FDI_DB_SCRIPT2.sql'.

```
Oracle SQL*Plus
File Edit Search Options Help
Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

Table created.

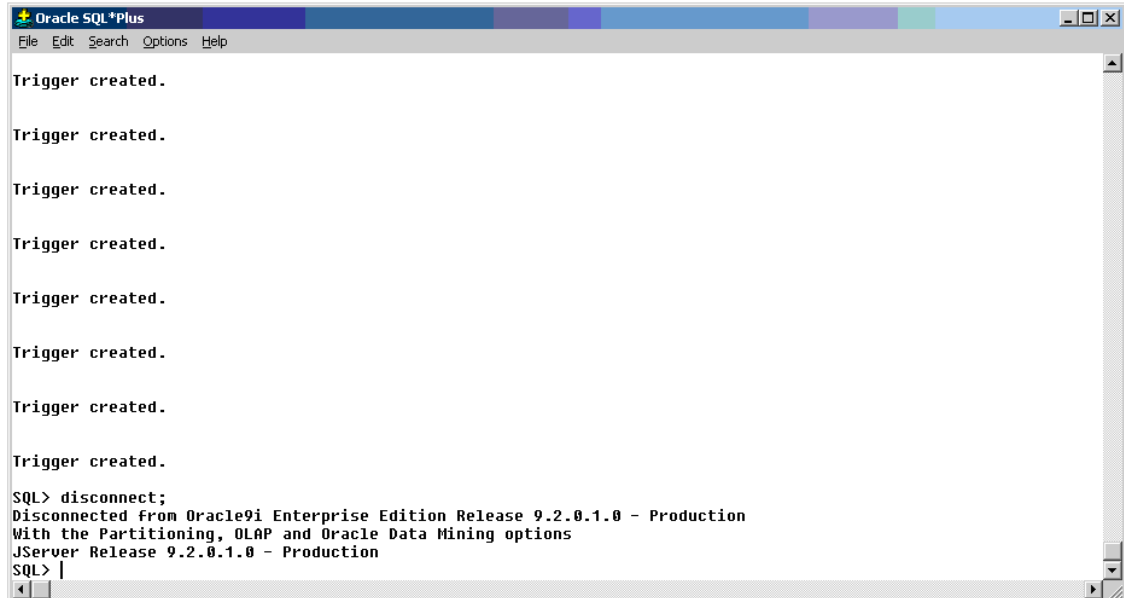
Table created.

SQL> connect fuego55wsfdi/password@gandalf
Connected.
SQL> @ e:\\fuego5.5\\j2ee\\FDI_DB_SCRIPT2.sql
```

Successful creation of all sequences and triggers should be followed by *Sequence created* and *Trigger created* messages. It is very important to verify all the sequences and triggers could be correctly created so that everything is properly set.

All these steps are the ones required from Oracle's Administrator. If for some reason, Oracle's Administrator needs to customize the SQL Scripts, these should be modified before the scripts are executed. These scripts should not be modified and re-executed once FuegoBPM Administrator continues to run the FDI Wizard.

After these steps, Oracle's Administrator can disconnect from SQL Plus as shown in the figure below.



The next step is for FuegoBPM Administrator to continue with FDI's Wizard. FuegoBPM Administrator should click on the *Next* button on the last FDI Wizard panel shown before. Then click on the *Close* button to finish the FuegoBPM Directory Service creation wizard. Successful creation should show the new FuegoBPM Directory Service in the Configuration's Panel.

After creating the Database, Oracle's Administrator should grant permissions to the FuegoBPM Directory Service Oracle User to access the DBA_PENDING_TRANSACTIONS table. This is needed in case transactions need to be rolled back when involving information persisted in the FuegoBPM Directory Service. The following statement can be executed with Oracle's SQL Plus:

```
GRANT SELECT ON DBA_PENDING_TRANSACTIONS TO Fuego55WSFDI;
```

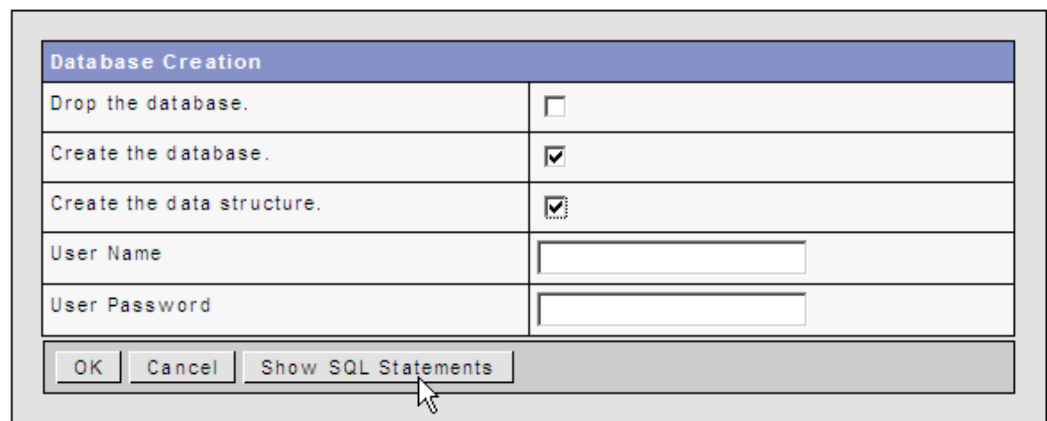
Chapter 12. Appendix B: Creating FuegoBPM Server Database using SQL-Plus

Creating a Server Database using an SQL script

To generate the script to create the Server database

1. Select the **Manage Database** option. The Database management pane appears.

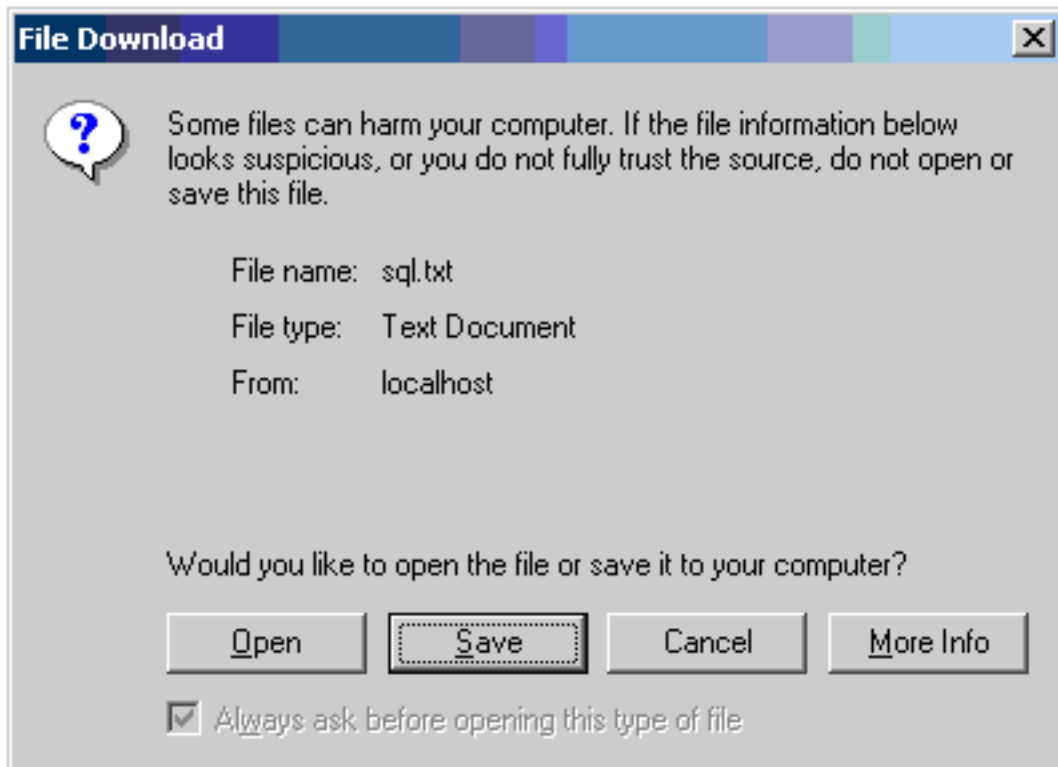
[Servers](#) > [Edit Server ServerDoc.](#) > [Manage Database](#)



Database Creation	
Drop the database.	<input type="checkbox"/>
Create the database.	<input checked="" type="checkbox"/>
Create the data structure.	<input checked="" type="checkbox"/>
User Name	<input type="text"/>
User Password	<input type="text"/>
<input type="button" value="OK"/> <input type="button" value="Cancel"/> <input type="button" value="Show SQL Statements"/>	

2. Select the **Create database** and **Create data structure** check boxes for the SQL Script to include the right information.

Click on the *Show SQL Statements* button. The following dialog will pop up asking to save the SQL Script with the Server Database SQL creation statements.



Next, you should provide the SQL Script to the Database's Administrator so the statements are executed.

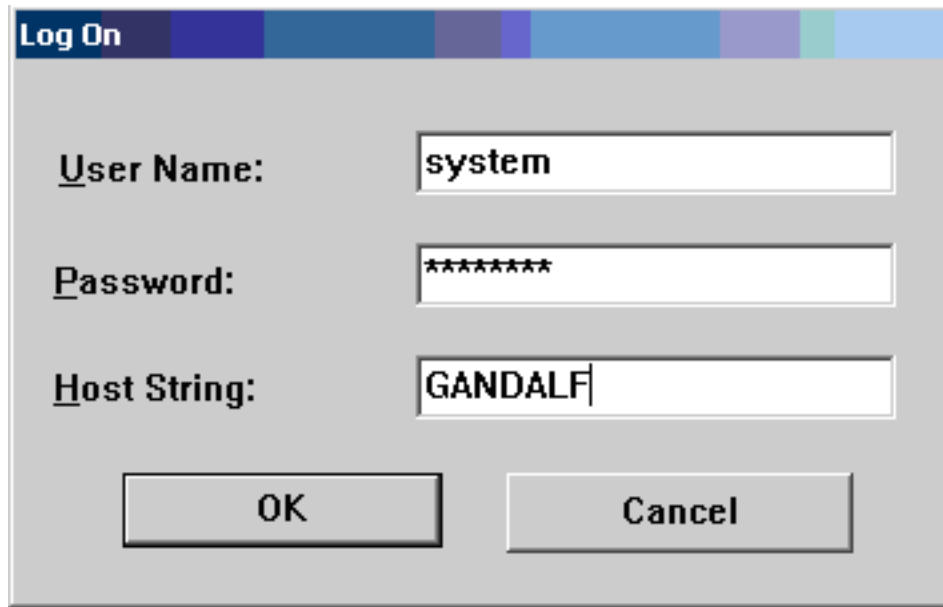
Running the Server DB SQL creation script in Oracle SQL-Plus

After saving the SQL script to create the FuegoBPM Server Database structure manually, you need the Oracle database Administrator to run this script.

The first line of the script should be modified by Oracle's Administrator so that a valid password is provided for the FuegoBPM Server Database Oracle User. The "*****" occurrence should be replaced with the actual password.

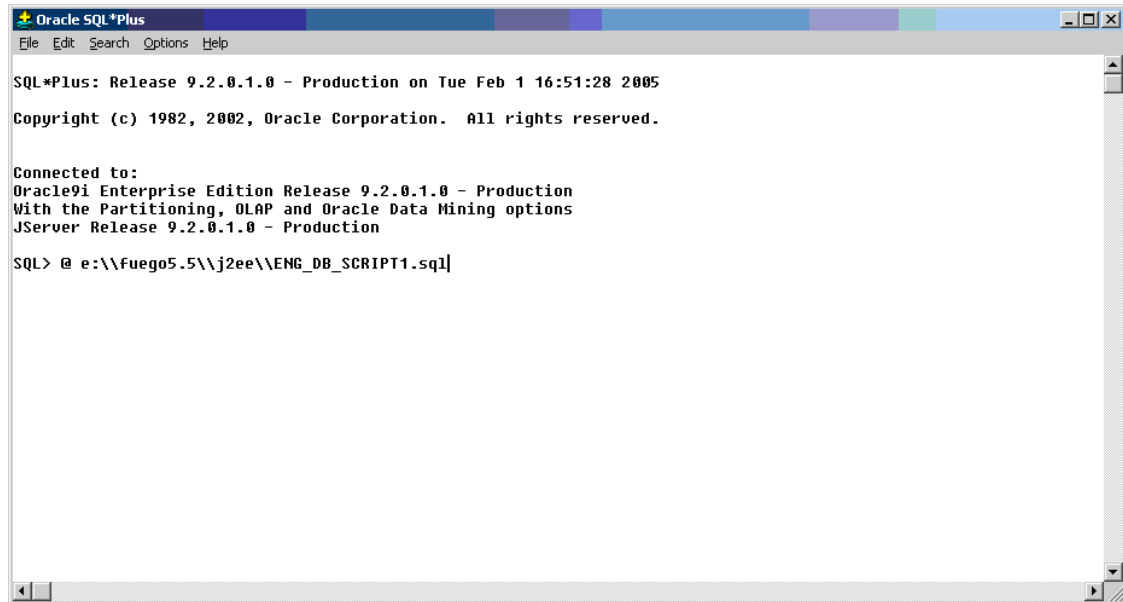
To proceed with the FuegoBPM Server Database creation, Oracle's user should connect to the Oracle Instance where the FuegoBPM Database will be created using SQL Plus. The Oracle user used to

execute this script should have enough permissions to create Oracle Users as well as creating tables and assigning connection and resource permissions. The SQL Script generated from FuegoBPM Web Console is prepared to work with this Editor and the syntax may not be well interpreted by other SQL Editors. The dialog below depicts the connection to the Oracle Instance using SQL Plus.



The image shows a 'Log On' dialog box with a title bar containing the text 'Log On' and a series of colored buttons. The dialog has three input fields: 'User Name:' with the value 'system', 'Password:' with the value '*****', and 'Host String:' with the value 'GANDALF'. At the bottom, there are two buttons: 'OK' and 'Cancel'.

Next, Oracle's Administrator should run the FuegoBPM Server Creation SQL Script as shown in the figure below.



Oracle's Administrator should confirm that all operations could be successfully executed to enforce a correct FuegoPM Server Database creation.

After successfully creating the Database, Oracle's Administrator should grant permissions to the FuegoBPM Server Oracle User to access the DBA_PENDING_TRANSACTIONS table. This is needed in case transactions need to be rolled back when the FuegoBPM Server cannot successfully complete a process activity task. The following statement can be executed with Oracle's SQL Plus:

```
GRANT SELECT ON DBA_PENDING_TRANSACTIONS TO Fuego55WASDB;
```

After completing these steps, the configuration for the FuegoBPM Server Database is completed.

Chapter 13. Appendix C: Tips

Tips when using Oracle 9i Database

If the XA support is not provided with the default Oracle Instance installation, you can run the following SQL scripts:

```
$ORACLE_HOME\javavm\install\initxa.sql  
$ORACLE_HOME\javavm\install\initjvm.sql
```

The initxa.sql script configures the database for XA. Once it runs successfully, your database is configured for XA. The script may run successfully the first time you try. Unfortunately, it probably will not run successfully because some of the database's memory spaces are too small. To fix this, run the initjvm.sql script. It will probably fail too, but in doing so it will indicate which parameters need to be adjusted.

Chapter 14. Appendix D: Configuring Oracle's JDBC Provider in WebSphere

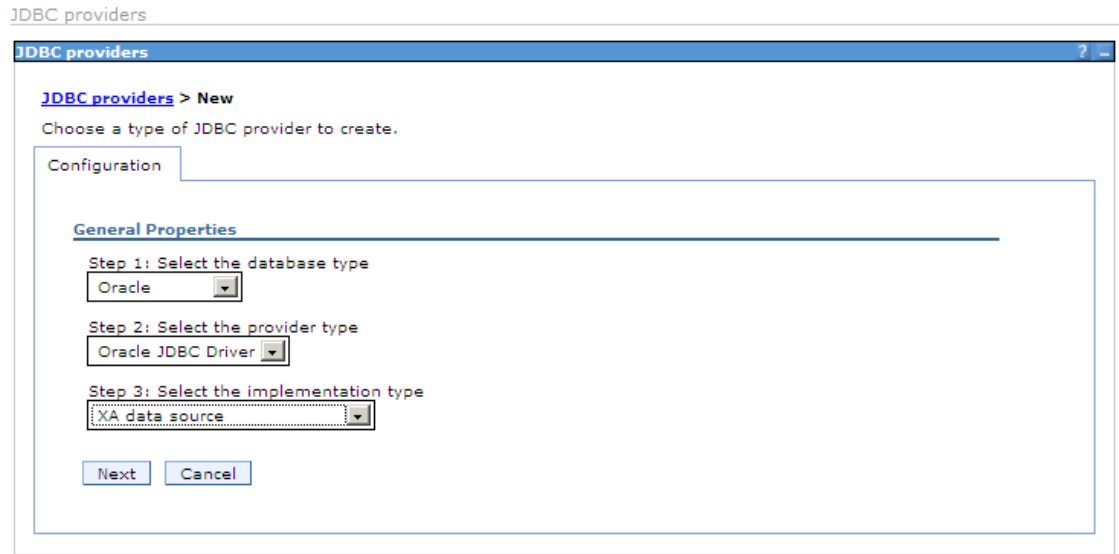
Configuring Oracle's JDBC Provider

Before the IBM's Administrator creates a **JDBC Data Source** for the *FuegoBPM Directory Service Database* already created and deployed on Oracle 9i RDBMS, the Oracle JDBC Provider has to be installed. If Oracle's JDBC drivers are already available for WebSphere, you can ignore these steps. Otherwise, check the instructions below. However, make sure you are using the latest versions of the JDBC drivers. For Oracle, FuegoBPM requires 10.2.0.1.0.

The WebSphere's Administrator should create a new directory in the WebSphere box and copy Oracle's 10g JDBC Driver into this directory.

The next step is to configure a new Oracle Provider. This can be achieved navigating the following entries in WebSphere's navigational tree on the left: *Resources -> JDBC Providers* and clicking on the *New* button.

A new panel is rendered where the WebSphere's Administrator should select Oracle XA JDBC Driver. It is important to remark that the XA version of the JDBC Driver is needed to enforce distributed transactions. The JDBC Driver selection is depicted in the figure below:



After selecting the appropriate Oracle JDBC driver, all the properties for this Oracle JDBC Provider are presented. In this case, you can accept the defaults proposed by WebSphere as shown in the figure below.

[JDBC providers](#) > New

JDBC providers are used by the installed applications to access data from databases.

Configuration

General Properties

* Scope
cells:ApoloNode01Cell:nodes:ApoloNode01

* Name
Oracle JDBC Driver (XA)

Description
Oracle JDBC Driver (XA)

Class path
\${ORACLE_JDBC_DRIVER_PATH}/ojdbc:

Native library path

* Implementation class name
oracle.jdbc.xa.client.OracleXADataSource

The additional properties will not be available until the general properties for this item are saved.

Additional Properties

- Data sources
- Data sources (Version 4)

The Class Path defaults to the *ORACLE_JDBC_DRIVER_PATH* environment variable. For further information on how to set it, refer to the following section in this appendix.

If the Administrator prefers not to set the variable, the absolute path can be defined.

Click on the *Ok* button to continue.

Setting up Oracle's JDBC Driver Environment Variable

After successfully copying Oracle's JDBC driver, you need to navigate

through WebSphere console through the following items in the navigational tree on the left panel: *Environment -> WebSphere Variables* and click on the *ORACLE_JDBC_DRIVER_PATH* entry as shown below. Define this path with the full path directory where the driver was copied to

Preferences

New Delete

☐ ☐ ☐ ☐

Select	Name	Value	Scope
<input type="checkbox"/>	APP_INSTALL_ROOT	\${USER_INSTALL_ROOT}/installedApps	cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	CLOUDSCAPE JDBC DRIVER_PATH	\${WAS_INSTALL_ROOT}/cloudscape/lib	cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	CONNECTJDBC JDBC DRIVER_PATH		cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	CONNECTOR_INSTALL_ROOT	\${USER_INSTALL_ROOT}/installedConnectors	cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	DB2390 JDBC DRIVER_PATH		cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	DB2UNIVERSAL JDBC DRIVER NATIVEPATH		cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	DB2UNIVERSAL JDBC DRIVER_PATH		cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	DB2 JDBC DRIVER_PATH		cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	DEPLOY_TOOL_ROOT	\${WAS_INSTALL_ROOT}/deploytool/itp	cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	DRIVER_PATH	\${WAS_INSTALL_ROOT}	cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	INFORMIX JDBC DRIVER_PATH		cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	JAVA_HOME	C:\was6\IBM\WebSphere\AppServer\java	cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	LOG_ROOT	\${USER_INSTALL_ROOT}/logs	cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	MQJMS_LIB_ROOT	\${MQ_INSTALL_ROOT}/java/lib	cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	MQ_INSTALL_ROOT	\${WAS_INSTALL_ROOT}/lib/WMQ	cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	MSSQLSERVER JDBC DRIVER_PATH		cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	ORACLE JDBC DRIVER_PATH		cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	OS400_NATIVE JDBC DRIVER_PATH		cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	OS400_TOOLBOX JDBC DRIVER_PATH		cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam
<input type="checkbox"/>	SYBASE JDBC DRIVER_PATH		cells:ApollonNode02Cell:nodes:ApollonNodeDocArTeam

Page: 1 of 2 Total 28

The following panel is shown where the location of the Oracle thin or oci JDBC driver needs to be specified. Click on the *OK* button to continue.

Appendix D: Configuring Oracle's JDBC Provider in WebSphere

[WebSphere Variables](#) > ORACLE_JDBC_DRIVER_PATH

Substitution variables allow specifying a level of indirection for values defined in the system, such as filesystem roots. Variables can be defined at the server, node, or cell level. When variables in different scopes have the same name, the order of resolution is server variables, then node variables, then cell variables.

Configuration

General Properties

* Name

ORACLE_JDBC_DRIVER_PATH

Value

c:\docdir\jdbclib

Description

The directory that contains the Oracle
thin or oci8 JDBC Driver.

Apply

OK

Reset

Cancel

Chapter 15. Appendix E: Undeploying and Redeploying a Project

Undeploying and Redeploying a Project in J2EE

Warning



Undeploying a process is not part of the regular procedure as all instances in the process are deleted. In a production environment is it highly discouraged.

To **undeploy** a project:

- 1) Undeploy the project from the webConsole (from the *Projects* panel)
- 2) Stop the project application from the Application server console.
- 3) Wait for complete undeployment in FuegoBPM WebConsole. Press the 'reload' button to check the status.
- 4) Uninstall the project from the Application Server console.

To **redeploy** a project that was once deployed but recently undeployed:

- 1) Deploy the project from the *Projects* panel.
- 2) Go to *J2EE Application Server files* and **regenerate** the ear file.
- 3) Deploy the project ear from the same panel.

To **deploy a new release** of a project (an updated project)

- 1) Publish and deploy it from FuegoBPM WebConsole in the *Projects* panel.
- 2) Go to *J2EE Application Server files* and regenerate the ear file.

- 3) Press the re-install button.
- 4) Start the project ear.

Chapter 16. Appendix F: Quick Cross Reference for Resources

Cross Reference for Resources in WebSphere

The following is a quick cross reference about the needed configuration in the WebSphere application server:

[Servers](#) > [Edit Server engine.](#)

Basic Configuration

Log

Execution

Services

Networking

Others

Application Server

Summary of the configuration needed in your websphere application Server

Server Runtime Datasource Lookup Name	XAengineDS
Server FDI Datasource Lookup Name	XAFDI DS
JMS Queue Name	queue/ToDoQueue
JMS Queue Connection Factory	XAConnectionFactory
JMS Topic Name for Server News	topic/EngineNews
JMS Topic Connection Factory	XATopicConnectionFactory
Listener bindings for message-driven beans	Activation Specification JNDI Name EngineActivSpec

RESOURCES on WEBSHERE'S APPLICATION SERVER

Resources -> JDBC Providers -> Oracle JDBC Driver (XA)
-> Data Sources

Resources -> JMS Providers -> Default Messaging ->

JMS Queue

JMS Queue Connection Factory

JMS Topic

JMS Topic Connection Factory

JMS Activation Specification