

# **Kuali HR 2.1.0 Installation Guide**

---

---

---

# Table of Contents

1. Overview .....	1
Suggested Operating Systems .....	1
Obtaining the Software .....	1
Installation Steps .....	1
2. Setup .....	3
Suggested Server Hardware .....	3
Installing and Configuring the Database Management System .....	4
Locations for Database Software .....	4
MySQL Database Preparation .....	5
Oracle Database Preparation .....	6
Suggested SQL Client Software .....	8
Install and Configure Required Software .....	8
Overview .....	8
Environment Variables .....	9
Java Development Kit (JDK) .....	10
Maven .....	11
Install and Setup Apache Tomcat .....	11
Copy JDBC drivers to Tomcat's lib directory .....	12
Install KPME software from distribution .....	12
Building the KPME Database .....	12
Load KPME Database .....	12
Configure KPME .....	14
Deploying the WAR file .....	14
Parameters .....	15

---

## List of Figures

2.1. Oracle XE admin webapp .....	7
-----------------------------------	---

---

## List of Tables

2.1. Locations for Database Software .....	4
2.2. Embedded Mode .....	15
2.3. KPME Database .....	15
2.4. Rice Database .....	16

---

# Chapter 1. Overview

KPME has the potential to run on most platforms that support a Java development environment (not simply a runtime environment), a servlet container, and an Oracle or MySQL relational database management system (RDBMS).

## Note

Only platforms and configurations that have been tested and are known to work with KPME are described within this guide. Other platforms and configurations may work, but have not been tested.

## Suggested Operating Systems

Since KPME is written in Java, it should in theory be able to run on any operating system that supports the required version of the Java runtime. However, it has been most actively tested on:

- Windows (7 and 8)
- Mac OS X (10.6+)
- Linux (Ubuntu)

Note that while Ubuntu Linux is the distribution most frequently used for testing, other Linux distributions such as Fedora, Red Hat Enterprise Linux, CentOS, Debian, Gentoo, Arch, and others should also be able to run KPME.

Additionally, KPME will likely work on other Unix operating systems such as Oracle Solaris and IBM AIX, although the software has not been tested here.

KPME has only been tested with the oracle distribution of the JDK. Any other jdk (IBM, OpenJDK, etc) must be used at your own risk.

## Obtaining the Software

1. **Download:** The KPME software can be downloaded from <http://kuali.org/kpme/download>
2. **Maven Repository** - <http://nexus.kuali.org/content/groups/public/>
3. **Subversion Repository** - <http://svn.kuali.org/repos/kpme/tags/kpme-2.1.0>

## Installation Steps

All KPME 2.1.0 installations follow the same core steps:

1. Install a database and JDBC drivers.
2. Install and configure a JDK and other required software.
3. Create the database schema and populate it.
4. Configure KPME software.

5. Test the installation

This Guide will provide installation instructions for the KPME server.

---

# Chapter 2. Setup

This chapter is designed to provide simple step-by-step instructions on how to set up a KPME server intended for enterprise deployment with an existing Rice server.

The steps to install and setup an embedded server with KPME are:

1. Determine your expected load and storage needs and consult the Suggested Server Hardware section for guidance. Install OS.
2. Install & configure the database management system.
3. Install & configure required software
4. Install and configure Tomcat.
5. Install and configure KPME.
6. Launch KPME.

## Note

Rather than install and setup as the root user on systems designed for production, you may want to create a non-privileged user named something like 'kpme' to use for this purpose.

## Suggested Server Hardware

Note that hardware needs may vary depending on the amount of expected load, the operating system being used, and the number of applications that are integrated with KPME. KPME is typically deployed as a client application connected to a separate Rice server, each with their own separate database deployed on separate database server from either application.

The recommended minimum requirements are as follows:

- Two processor cores of a recent CPU architecture along with 2 GB of reserved RAM. Minimum hardware and settings for the database should be based on vendor recommendations, but there should be high bandwidth with low latency between the web application and database as this is a database intensive application.
- At least 1 GB of heap space (-Xmx1g) and 256 MB of permanent generation space (-XX:MaxPermSize=256m) reserved for the application to start and run. For production use, at least 2 GB of heap and 512 MB of permanent generation space are recommended.

## Note

Other recommended settings are -server (to improve resource utilization for long-running tasks) and -XX:+UseConcMarkSweepGC (to improve garbage collection performance).

- 100 Mbit/s network card (gigabit preferred)
- 500 MB of free hard disk space (for Tomcat server and web application)



## Note

Additional space needed if storing attachments or not clearing out log files.

# Installing and Configuring the Database Management System

KPME was developed using two relational database management systems: **MySQL** and **Oracle**. The typical production install involves running the KPME application separate from the database server, however both can be run on the same machine for development purposes.

KPME runs, and has been tested with the following versions:

- Oracle
  - Oracle Express Edition (XE)

Use the Oracle JDBC Driver to connect to these databases.

Enterprise Oracle installations like 10g or 11g should work with KPME but they have not been tested. Ensure that the Oracle database you intend to use encodes character data in a UTF variant by default. For Oracle XE, this entails downloading the "Universal" flavor of the binary, which uses AL32UTF8.

- MySQL
  - MySQL 5.1.+

Use the MySQL Connector/J (5.1.+) to connect to MySQL databases.

You should be able to adapt KPME to other standard relational databases (e.g., Sybase, Microsoft SQL Server, DB2, etc.). However, this Installation Guide does not provide information for running KPME with these products.

## Locations for Database Software

Below are locations from which Oracle and MySQL could be downloaded at the time of release of Rice 2.1.0.

**Table 2.1. Locations for Database Software**

Software	Download Location
Oracle Standard and Enterprise Editions	<a href="http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html">http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html</a>
Oracle Express Edition	<a href="http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html">http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html</a>
Oracle JDBC Driver	<a href="http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html">http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html</a>
MySQL	<a href="http://dev.mysql.com/downloads/">http://dev.mysql.com/downloads/</a>

Software	Download Location
MySQL Connector/J JDBC Driver	<a href="http://www.mysql.com/products/connector/">http://www.mysql.com/products/connector/</a>

## MySQL Database Preparation

KPME supports both MySQL and Oracle databases. However, MySQL is easier to install than Oracle and uses less machine resources so many developers prefer to use that when getting started with KPME.

### Installation

The installation steps for MySQL are going to be different for each platform. Please download the latest version MySQL Server from the location listed in the [Locations for Database Software](#) section of this document and follow the installation instructions for your platform.

#### Note

You may be required to create an account on the MySQL site in order to download the software.

Please be sure to follow the instructions for installing MySQL on your platform very carefully. If downloading for Mac OS X, **be careful** to download the appropriate version for your platform (32-bit vs. 64-bit and 10.6 vs 10.7)

### Configuration

There are a few MySQL database configuration options that are required in order for KPME to work properly. These will need to be set in either your **my.cnf** or **my.ini** file. The location and names of these files will differ depending on which platform you are working on. For details on where these files can be found, see the following document:

<http://dev.mysql.com/doc/refman/5.1/en/option-files.html>

Once you have located this file, please add the following options, paying special attention to the line that needs to be commented out:

```
[mysqld]
max_allowed_packet=20M
transaction-isolation=READ-COMMITTED
lower_case_table_names=1
max_connections=1000
innodb_locks_unsafe_for_binlog=1
...
# Be sure to comment this out if it's in the file!!!
#log-bin=mysql-bin
```

Note that the **[mysqld]** section may already be in your **my.cnf** file. If so, you can just add the options listed above underneath that section

#### Caution

It is very important that you comment out **log-bin**. Otherwise you will end up with some very bad problems later!

## Verification

Before verifying your mysql installation you will need to ensure that MySQL is running. Some of the platform-specific packages will set this up automatically (or allow you to install yourself in the case of Mac OS X). If MySQL is not starting automatically you can start it using a command like the following example from Mac OS X:

```
sudo /usr/local/mysql/bin/mysqld_safe
```

This will start the MySQL server.

To verify that you can actually connect to the server, execute the following at the command line:

```
mysql -u root -p
```

This should bring you to a command line client interface for the MySQL server. Type "show databases;" and press return. You should see output similar to the following.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.50-log MySQL Community Server (GPL)
```

```
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| test                     |
+-----+
3 rows in set (0.13 sec)

mysql>
```

## Install the JDBC Driver

KPME uses the MySQL Connector/J product as the native JDBC driver. When needed for building the project, Maven will automatically download this driver, so no additional action should be necessary.

## Oracle Database Preparation

### Installation

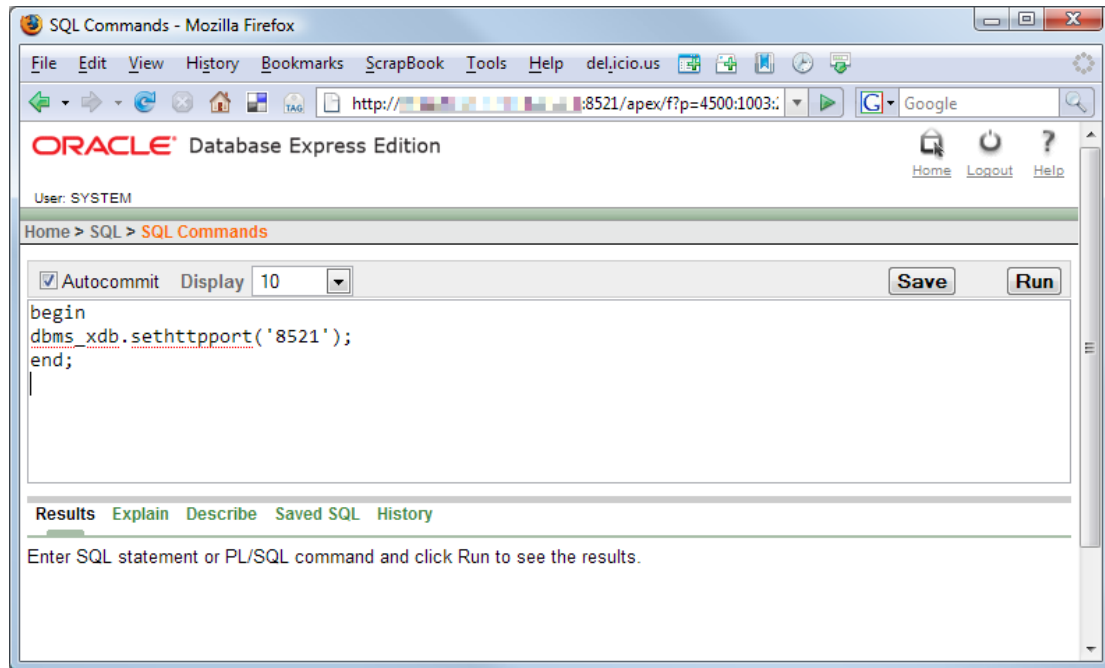
The installation steps for Oracle are going to be different for each platform and version of Oracle. Please download from the location listed in the [Locations for Database Software](#) section of this document and follow the installation instructions for your platform.

To run the database completely on your local machine, we recommend installing Oracle Express (XE). Please refer to the [Locations for Database Software](#) section of this Installation Guide to find the download location for this software.

## Configuration

By default, OracleXE registers a web user interface on port 8080. This is the same port that the development version of KPME is preconfigured to use. To avoid a port conflict, you must change the port that the OracleXE web user interface uses with the Oracle XE admin webapp:

**Figure 2.1. Oracle XE admin webapp**



If you prefer, you can use the Oracle SQL tool described here to change the OracleXE web user interface port: <http://daust.blogspot.com/2006/01/xe-changing-default-http-port.html>

Please edit your hosts file with an entry to refer to your Oracle database. When this Installation Guide refers to the Oracle database host server, it will be referred to in the examples as koracle.

Now edit the hosts file and add this:

```
<ip address of mysql server> koracle
```

## Verification

Before verifying your oracle installation you will need to ensure that Oracle is running. Some of the platform-specific packages will set this up automatically. If Oracle is not starting automatically you can start it using a command like the following example from Ubuntu:

```
sudo /etc/init.d/oracle start
```

This will start the Oracle server.

To verify that you can actually connect to the server, execute the following at the command line:

```
sqlplus
```

This should bring you to a command line client interface for the Oracle server.

SQL\*Plus: Release 11.2.0.4.0 - Production on Mon Jul 14 14:11:23 2014

Copyright (c) 1982, 2013, Oracle. All rights reserved.

SQL>

## Install the JDBC Driver

KPME uses the standard Oracle JDBC driver as the native JDBC driver.

1. Please download this driver from the location specified in the [Locations for Database Software](#) section of this Guide.
2. Copy the jdbc driver to the "LIB" directory of your application server.
3. If you do not already have Maven 3 installed, install it now using the instructions in the [Maven](#) section.
4. Use the [maven-install-plugin](#) to copy ojdbc14.jar into your local maven repository.

```
mvn install:install-file -DgroupId=com.oracle -DartifactId=ojdbc14 \
-Dversion=10.2.0.3.0 -Dpackaging=jar \
-Dfile=ojdbc14.jar
```

## Suggested SQL Client Software

To examine and test your database setup, SQL client software is useful. Any SQL client software that will connect to a MySQL or Oracle database will work. Two tools used by the development team are the MySQL Workbench and Oracle SQL Developer.

### MySQL Workbench

MySQL Workbench only works with MySQL and can be installed to connect to any running MySQL server. You can download and install it from the following URL:

- <http://www.mysql.com/downloads/workbench/>

### Oracle SQL Developer

Oracle SQL Developer is a Java-based program mainly used to connect to Oracle databases (but can connect to MySQL databases with a plugin). You can download and install it from the following URL:

- <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>

## Install and Configure Required Software

### Overview

KPME requires the following software to be setup and configured:

- Oracle Java Development Kit (JDK 1.7.x)

## Warning

You must use a JDK and not a Java runtime environment (JRE); the JDK you use must be either version 1.7.x or newer. Additionally, KPME has not been tested on JDKs other than Oracle. So alternative implementations like OpenJDK should be used at your own risk.

- Maven 3

## Environment Variables

First, some environment variables need to be configured.

### Mac OS X

Environment variables in Mac OS X can be set in a number of ways, but here we will show how to modify or create the .profile files in your user home directory. On OS X your user home directory is typically located at /Users/<username>.

An example .profile can be found below:

```
M2_HOME=/usr/local/maven
MAVEN_OPTS="-Xmx1024m -XX:MaxPermSize=768m"
JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/Versions/1.6/Home
CATALINA_HOME=/usr/local/tomcat
MYSQL_HOME=/usr/local/mysql

PATH="$M2_HOME/bin:$CATALINA_HOME:$MYSQL_HOME/bin:$PATH"
export PATH M2_HOME MAVEN_OPTS JAVA_HOME CATALINA_HOME MYSQL_HOME
```

### Note

It is important to export your environment variables once they are defined as the file above does.

### Windows XP

To get to the screen where you can define environment variables on Windows XP follow these steps:

1. Click on the "Start" button in the bottom left-hand corner.
2. On the resulting screen, right click on "My Computer".
3. In the context menu, click on "Properties".
4. This will open up the "System Properties" dialog window.
5. Click on the "Advanced" tab.
6. Click on the "Environment Variables" button.
7. You will see the screen where you can edit existing environment variables or define new ones.

### Note

The windows command line console must be closed and reopened in order for environment variable changes to be effective.

## Windows Vista and Windows 7

To get to the screen where you can define environment variables on Windows Vista or Windows 7 follow these steps:

1. Click on the "Start" button in the bottom left-hand corner.
2. On the resulting screen, right click on "Computer".
3. In the context menu, click on "Properties".
4. This will open up the Control Panel "System" dialog.
5. Click on the "Advanced system settings".
6. In the resulting window, click on the "Environment Variables..." button.
7. You will see the screen where you can edit existing environment variables or define new ones.

### Note

The windows command line console must be closed and reopened in order for environment variable changes to be effective.

## Java Development Kit (JDK)

### Installation

You should download and install the latest version of JDK 7. You can download it from the following URL: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

### Configuration

You will also want to set up your **JAVA\_HOME** environment variable to point to the installation directory of your JDK. In both Windows and Mac environments, the **java** executable program should already be on your path. But if it is not, you will want to include **JAVA\_HOME/bin** in your **PATH** environment variable.

If you do not know how to do this, see the **Environment Variables** section above for your platform.

### Verification

In order to verify that your JDK has been installed successfully, open a command prompt and type the following:

```
java -version
```

You should see output similar to the following:

```
java version "1.7.0_60"  
       Java(TM) SE Runtime Environment (build 1.7.0_60-b19)  
       Java HotSpot(TM) 64-Bit Server VM (build 24.55-b03, mixed mode)
```

If you receive an error indicating that the "java" command could not be found, please ensure that the java command is on your machine's **PATH** environment variable.

To prevent potential out of memory errors when running KPME, you should set your **JAVA\_OPTS** environment to a value like the following:

```
JAVA_OPTS="-Xmx1024m -XX:MaxPermSize=768m"
```

If you do not know how to do this, see the [Environment Variables](#) section above for your platform.

## Maven

Maven is the primary build tool used by the Kuali Rice project. Maven is based on a project object model (POM) that defines various standards and conventions surrounding the organization of a project. This facilitates a set of standard build goals and lifecycle phases (such as compile, test, package, etc.).

## Installation

To download version 3 of Maven, use the following link: <http://maven.apache.org/download.html>

Once you have downloaded the zip file, unzip it to a location of your choosing.

## Configuration

You will want to set your **M2\_HOME** environment variable to point to the location where you unzipped Maven. You will additionally want to include **M2\_HOME/bin** in your **PATH** environment variable so that maven can be executed from the command line without having to specify the full path.

Finally, to prevent potential out of memory errors when compiling KPME with Maven, you should set your **MAVEN\_OPTS** environment to a value like the following:

```
MAVEN_OPTS="-Xmx1024m -XX:MaxPermSize=768m"
```

If you do not know how to do this, see the [Environment Variables](#) section above for your platform.

## Verification

In order to verify that Maven has been installed successfully and is available on the path, open a command prompt and type the following:

```
mvn -version
```

You should see output like the following:

```
Apache Maven 3.0.3 (r1075438; 2011-02-28 10:31:09-0700)
Maven home: /usr/local/maven
Java version: 1.7.0_60, vendor: Apple Inc.
Java home: /System/Library/Java/JavaVirtualMachines/1.7.0.jdk/Contents/Home
Default locale: en_US, platform encoding: MacRoman
OS name: "mac os x", version: "10.7.2", arch: "x86_64", family: "mac"
```

If you receive an error indicating that the "mvn" command could not be found, please ensure that the directory that includes the mvn executable (**M2\_HOME/bin**) is on your machine's **PATH** environment variable.

## Install and Setup Apache Tomcat

KPME 2.1.0 supports the following Tomcat versions:



- Tomcat 6

Tomcat 7

Tomcat 8

Please visit the [Apache Tomcat site](#) for information on how to install and configure Tomcat.

Other servlet containers can be used with KPME, but this guide will focus on Tomcat.

## Copy JDBC drivers to Tomcat's lib directory

Copy the JDBC drivers downloaded earlier ([Locations for Database Software](#)) and paste them into the lib folder in your Tomcat directory.

## Install KPME software from distribution

The quickest way to get KPME installed is to download the binary distribution from the [kuali.org download site](#). Once downloaded, decompress the software.

# Building the KPME Database

## Load KPME Database

The KPME database release scripts contain all the SQL commands needed to install a new schema for version 2.1.0 database using Rice embedded mode with the database objects (tables, constraints, bootstrap data) for the KPME application.

### Note

You will need to already have a working instance of a Rice 2.3.6 up and running since you will need to run sql statements against that Rice database. Before running any scripts against the rice database, please update your existing rice installation to Rice 2.3.6.

## Prepare a KPME MySQL Database

1. Create a MySQL user with a username of less than 8 characters.
2. Create a MySQL default schema for the user with the default character set of UTF8. If a different character set is desired, the scripts will need to be updated to the new character set.
3. Give the new MySQL user the following privileges on the schema:
  - SELECT
  - INSERT
  - UPDATE
  - DELETE
  - CREATE
  - DROP

- INDEX
- ALTER
- CREATE\_VIEW
- CREATE\_ROUTINE
- ALTER\_ROUTINE
- CREATE\_TMP\_TABLE
- LOCK\_TABLES

## Prepare a KPME Oracle Database

1. Create an Oracle user with a username of less than 8 characters.
2. Give the new Oracle user the following privileges:
  - DEFAULT TABLESPACE set to <Users Tablespace> (the intended location where the schema database structures and base bootstrap data are stored).
  - QUOTA UNLIMITED set to <Users Tablespace>
  - CREATE SESSION
  - CREATE SYNONYM
  - CREATE PROCEDURE
  - CREATE TRIGGER
  - CREATE TABLE
  - CREATE TYPE
  - CREATE VIEW
  - CREATE SEQUENCE

### Note

A user's DEFAULT TABLESPACE is set with the CREATE USER statement or ALTER USER statement. The TABLESPACE should not be the SYSTEM tablespace.

## Populate the KPME Database

Change to the directory **db-scripts** in the distribution. Run the scripts in order for the applicable database type. Client scripts run on the KPME database and server scripts run on the existing Rice database. The server-install scripts assume that an existing rice database of version 2.3.6 or newer is being used.

### Full Install

- 1. **2.1.0/oracle/kpme-2.1.0-oracle-client-install.sql**
- Oracle Server

1. **2.1.0/oracle/kpme-2.1.0-oracle-server-install.sql**
- 1. **2.1.0/mysql/kpme-2.1.0-mysql-client-install.sql**
- MySQL Server
1. **2.1.0/mysql/kpme-2.1.0-mysql-server-install.sql**

### Upgrade from 2.0.1, 2.0.2, or 2.0.3

- Oracle Client
- 1. **2.1.0/oracle/kpme-2.1.0-oracle-client-upgrade.sql**
- Oracle Server
- 1. **2.1.0/oracle/kpme-2.1.0-server-client-upgrade.sql**
- MySQL Client
- 1. **2.1.0/mysql/kpme-2.1.0-mysql-client-upgrade.sql**
- MySQL Server
- 1. **2.1.0/mysql/kpme-2.1.0-mysql-server-upgrade.sql**

## Verifying your Database Installation

At this point, your KPME database should be successfully installed. To verify this, log into your database and verify that tables exist.

# Configure KPME

## Deploying the WAR file

1. Change to the **binary** folder in the distribution. Copy the **kpme-web-2.1.0.war** file to the directory that contains web applications in your servlet container. For Tomcat, this is **<Tomcat-root-directory>/webapps**.
2. Copy the database-specific JDBC driver to the **<Tomcat-root-directory>/lib**.
3. Place **kpme-config.xml** in the **<userhome>/kuali/main/dev/** directory. Change the "dev" to the environment name you are using if not dev.
4. Configure the **kpme-config.xml** file and edit parameters as desired.
  - You will at least need to fill in the parameters listed in the [Parameters](#) section.
  - Other commonly changed parameters are inherited from Rice and are normally changed to reflect the local settings for KPME: **application.url**, **app.context.name**, **log4j.settings.path**, **log4j.settings.reloadInterval**, **keystore.alias**, **keystore.file**, **keystore.password**, and **encryption.key**. Please see the Rice documentation for more information.
5. Startup Tomcat to deploy KPME. Watch the logs to ensure that it started up successfully.

6. The KPME application should now be accessible at **http://localhost:8080/kpme-dev**.

7. Login as 'admin.' Ingest the KPME KEW documents.

- Click the System Admin tab and then select XML Ingester from the Workflow pane.
- Change to the folder **db-scripts/workflow** in the distribution and ingest all files in that directory. You may alternatively put all of these files (including folders) into one zip file and just ingest the one zip file if it is easier. Ingestion can take several minutes, during which time the activity indicator on your browser will just stay animated.

## Parameters

The tables below have the basic set of parameters for **kpme-config.xml** that you need to get an instance of KPME running. Please use these tables as a beginning reference to modify your **kpme-config.xml** file.

**Table 2.2. Embedded Mode**

Parameter	Description	Examples or Values
rice.server.url	The external URL used to access the Rice instance.	http://localhost:9080/kr-dev
dev.mode	Whether or not the server is running in development mode. Use false for production.	false
kim.mode	The mode that KIM is running in. Use EMBEDDED for embedded mode.	EMBEDDED
kcb.mode	The mode that KCB is running in. Use EMBEDDED for embedded mode.	EMBEDDED
kew.mode	The mode that KEW is running in. Use EMBEDDED for embedded mode.	EMBEDDED
ken.mode	The mode that KIM is running in. Use REMOTE for embedded mode.	REMOTE
ksb.mode	The mode that KSB is running in. Use REMOTE for embedded mode.	REMOTE
kns.mode	The mode that KNS is running in. Use LOCAL even in embedded mode.	LOCAL
krms.mode	The mode that KRMS is running in. Use REMOTE for embedded mode.	REMOTE
coreservice.mode	The mode that the core service is running in. Use REMOTE for embedded mode.	REMOTE
location.mode	The mode that the location module is running in. Use REMOTE for embedded mode.	REMOTE
core.mode	The mode that the core application is running in. Use LOCAL even in embedded mode.	LOCAL

**Table 2.3. KPME Database**

Parameter	Description	Examples or Values
datasource.obj.platform	OJB platform name	Oracle9i or MySQL
datasource.username	Username for the database	

---

Parameter	Description	Examples or Values
datasource.password	Password for the database	
datasource.url	JDBC URL to the database	<ul style="list-style-type: none"><li>• jdbc:mysql://localhost:3306/kpmedev</li><li>• jdbc:oracle:thin:@localhost:1521:XE</li></ul>

**Table 2.4. Rice Database**

Parameter	Description	Examples or Values
server.datasource.ojb.platform	OJB platform name	Oracle9i or MySQL
server.datasource.username	Username for the database	
server.datasource.password	Password for the database	
server.datasource.url	JDBC URL to the database	<ul style="list-style-type: none"><li>• jdbc:mysql://localhost:3306/krdev</li><li>• jdbc:oracle:thin:@localhost:1521:XE</li></ul>