



Jive Interactive Intranet

## 9.x Administrator Guide

### Database Configuration

# Notices

---

For details, see the following topics:

- [Notices](#)
- [Third-party acknowledgments](#)

## Notices

Copyright © 2000–2021. Aurea Software, Inc. (“Aurea”). All Rights Reserved. These materials and all Aurea products are copyrighted and all rights are reserved by Aurea.

This document is proprietary and confidential to Aurea and is available only under a valid non-disclosure agreement. No part of this document may be disclosed in any manner to a third party without the prior written consent of Aurea. The information in these materials is for informational purposes only and Aurea assumes no responsibility for any errors that may appear therein. Aurea reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of Aurea to notify any person of such revisions or changes.

You are hereby placed on notice that the software, its related technology and services may be covered by one or more United States (“US”) and non-US patents. A listing that associates patented and patent-pending products included in the software, software updates, their related technology and services with one or more patent numbers is available for you and the general public’s access at <https://markings.ip-dynamics.ai/esw/> (the “Patent Notice”) without charge. The association of products-to-patent numbers at the Patent Notice may not be an exclusive listing of associations, and other unlisted patents or pending patents may also be associated with the products. Likewise, the patents or pending patents may also be associated with unlisted products. You agree to regularly review the products-to-patent number(s) association at the Patent Notice to check for updates.

Aurea and Aurea Software are registered trademarks of Aurea Software, Inc. in the United States and/or other countries. Additional Aurea trademarks, including registered trademarks, are available at: <https://www.aurea.com/legal/trademarks/>. Jive is a registered trademark of Jive Software, Inc. in the United States and/or other countries. Additional Jive trademarks, including registered trademarks, are available at: <https://www.jivesoftware.com/legal/>.

---

## Third-party acknowledgments

The following third-party trademarks may appear in one or more Jive guides:

- Amazon is a registered trademark of Amazon Technologies, Inc.
- Apache and Derby is a trademark of Apache Software Foundation.
- Chrome is a trademark of Google Inc.
- Eclipse is a registered trademark of the Eclipse Foundation, Inc.
- HP-UX is a registered trademark of Hewlett-Packard Development Company, L.P.
- IBM, AIX, DB2, and WebSphere are registered trademarks of International Business Machines Corporation.
- Intel and Pentium are registered trademarks of Intel Corporation in the U.S. and/or other countries.
- JBoss is a registered trademark, and CentOS is a trademark, of Red Hat, Inc. in the U.S. and other countries.
- Linux is a registered trademark of Linus Torvalds.
- Microsoft, Active Directory, Internet Explorer, SharePoint, SQL Server, Visual Studio, and Windows are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Mozilla and Firefox are registered trademarks of the Mozilla Foundation.
- Oracle and Java are registered trademarks of Oracle and/or its affiliates.
- Progress and OpenEdge are registered trademarks of Progress Software Corporation or one of its subsidiaries or affiliates in the U.S. and other countries.
- Red Hat and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the U.S. and other countries.
- SAP and SAP NetWeaver are registered trademarks of SAP SE in Germany and in several other countries.
- SUSE is a registered trademark of SUSE, LLC.
- Ubuntu is a registered trademark of Canonical Limited in the United States and/or other countries.
- UNIX is a registered trademark of The Open Group in the United States and other countries.

All other marks contained herein are for informational purposes only and may be trademarks of their respective owners.

---

---

# Table of Contents

<b>Aurea global support.....</b>	<b>5</b>
<b>Chapter 1: Database configuration and best practices.....</b>	<b>6</b>
Database configuration.....	6
Supported database engines.....	8
Setting up new databases.....	9
Upgrading databases.....	10
Postgres database best practices.....	11
Oracle database best practices.....	12
Oracle database drivers and configurations.....	13
Additional configuration for using Oracle 12 and 19 databases with SSL/TLS connection .....	14
Microsoft SQL server database best practices.....	17
MySQL database best practices.....	18
Quick database setup for evaluations.....	19

# Aurea global support

---

If you encounter a problem while using an Aurea product or require assistance with downloading the software or upgrading a product release, please, try to:

- Search the articles on the [Aurea Knowledge Base](#) for solutions to your issues.
- Search the product documentation and other product-related information that are also available on [Support Central](#).

If you still cannot find a solution, open a ticket on [Aurea Support Central](#). Information about the support organization is available on [Support Portal](#) as well.

You can also find the setup files on [Support Portal](#).

For information about purchasing an upgrade or professional services, contact your account executive. If you do not know who your account executive is, or for other queries, contact us through our [website](#).

# 1

## Database configuration and best practices

---

*Make sure your databases are ready before you install or upgrade Jive.*

For details, see the following topics:

- [Database configuration](#)
- [Supported database engines](#)
- [Setting up new databases](#)
- [Upgrading databases](#)
- [Postgres database best practices](#)
- [Oracle database best practices](#)
- [Oracle database drivers and configurations](#)
- [Additional configuration for using Oracle 12 and 19 databases with SSL/TLS connection](#)
- [Microsoft SQL server database best practices](#)
- [MySQL database best practices](#)
- [Quick database setup for evaluations](#)

### Database configuration

Installation or upgrade goes more smoothly if you complete the database-related tasks first. Refer to the database-specific best practices below for information about how to configure your database to work with Jive.

#### Prerequisites

To install Jive, you need to set up three databases: one each for the core application, the Activity Engine, and Analytics. While it's possible to run Jive without Analytics, any substantial installation needs to include it.

For new installations, the database and application users must be created before installation. You should also follow the recommended best practices for the database vendor and version selected.

#### Databases administration

We highly recommend having a skilled database administrator in charge of your Jive databases.

## Placement of your databases on separate servers

For large installations, it is recommended that the core application database, the Activity Engine database, and the Analytics database be hosted on separate database engines. At a minimum, disk and memory resources should be isolated for each service.

## Why separate the Core Application and the Analytics databases?

The following points explain why it's especially important for your Analytics database to be separate from the application database. Ideally, they should be stored on two separate servers. While we technically support storing the Analytics and application databases on one server, here's why it's important to ensure isolation:

- Analytics has a very different access pattern from the Core Application database: many streaming writes, occasional reads, and busy activity once a day when the ETL runs.
- The storage requirements for Analytics are very different from the Core Application database requirements. The physical size of the Analytics databases grows much larger than the application database on busy systems. The IO profiles (the application database is a typical OLTP load, the Analytics database is OLAP) and the usage profiles (potentially large result sets, aggregates and ad-hoc queries are expected on the Analytics database) also support isolation of the application and analytics resources.
- The Core Application database is more critical than the Analytics database. Therefore, the Analytics database should not be in a position to demand resources that the core database may need.
- The Analytics database is intended to be accessed by users running ad-hoc queries. Because the application cannot predict which queries users may run, it is possible that a bad query could execute and thus consume all of the server's capacity.

## Attachment storage

You should configure attachments for storage by using a file system provider rather than the database. Storing attachments in the database is possible, but is known to cause severe performance issues when the document store gets large and contends with the database cache. For more information about how to set up your file system storage, see [Required external components](#) and [Configuring binary storage provider](#).

## Database collation

The database or database instance must be configured for UTF8 collation. Refer to the instructions for your individual database platform for instructions.

## Supported database engines

Jive supports recent versions of several databases.

The following database platforms are supported. For more information about setting them up to deploy Jive, see the [Database configuration and best practices](#) on page 6 section.

**Jive Core and Activity Engine databases**

- Jive 9.7.0 and later
  - Oracle 11gR2 1, 12c 1, 19c 1
  - PostgreSQL 9.4, 9.5, 9.6, 11, 12.6
  - MySQL 5.6, 5.7
  - MS SQL Server 2012, 2014
- Jive 9.4.0–9.6.0:
  - Oracle 11gR2 1, 12c 1, 19c 1
  - PostgreSQL 9.4, 9.5, 9.6, 11
  - MySQL 5.6, 5.7
  - MS SQL Server 2012, 2014
- Jive 9.0.0–9.3.0:
  - Oracle 11gR2 1, 12c 1
  - PostgreSQL 9.4, 9.5, 9.6. Version 9.6 is recommended
  - MySQL 5.6, 5.7
  - MS SQL Server 2012, 2014

**Jive Analytics (CommunityManager Reports) database**

- Jive 9.7.0 and later:
  - Oracle 11gR2 1, 12c 1, 19c 1
  - PostgreSQL 9.0, 9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 11, 12.6
- Jive 9.4.0–9.6.0:
  - Oracle 11gR2 1, 12c 1, 19c 1
  - PostgreSQL 9.0, 9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 11
- Jive 9.0.0–9.3.0:
  - Oracle 11gR2 1, 12c 1
  - PostgreSQL 9.0, 9.1, 9.2, 9.3, 9.4, 9.5, 9.6. Version 9.6 is recommended

## Setting up new databases

When setting up new databases, you should ensure sufficient isolation, and set an appropriate number of connections.

During the installation process, you need to create up to three databases: the Core Application database, the Activity Engine database, and (optionally) the Analytics database. These systems should be installed on separate, isolated resources, as described in [Database configuration](#) on page 6. Although you may be tempted to ignore this recommendation if you have a robust, high-capacity system, you should still seriously consider IO and memory isolation. It's also very important to maintain isolation of security roles.

---

**Note:** Refer to the system-specific best practices for your database before installing.

---

The maximum number of incoming connections must be set to accommodate the sum of max connections across all web servers, plus a small administrative overhead.

For example, if 5 application nodes are set to from 50 to 150 max connections, the database should be able to allow as many as  $(5 \times 150) = 750$  connections from the application DB, +n for administrative connections should all connections be in use. A typical overage is +5–10 connections.

Analytics databases only require connections for the application server listeners and administrative users. Typically, the maximum connections number is much lower than the application database requires. The Activity Engine database default is a maximum of 50 connections. This typically does not need to be increased.

## Upgrading databases

If you plan to leave the database on the same system, no special configuration is required. If you're moving to a new database engine or system, migrate the system first, then upgrade.

If you're upgrading to a version whose database schema is different from the previous version, the application automatically detects the difference after you've upgraded. When you next start the application and go to the Admin Console, you will be prompted to start the database upgrade. This is a necessary step before you can start using the application after an upgrade. You don't need to run any database-related upgrade scripts unless you are moving to a different database engine.

To upgrade the database:

1. Shut down all the services.
2. If you're using the same database engine (for example, moving SQL Server to SQL Server), back up and restore the database to the new instance.
3. If you're using a different database engine after the upgrade (for example, moving Oracle to PostgreSQL), use scripts and ETL tools to build the database and transfer the data. For more information about the properties you need to pay attention to when migrating a database, see [Restoring database with persistent properties](#).

---

<sup>1</sup> Additional configuration required. For more information, see [Oracle database drivers and configurations](#) on page 13 and [Additional configuration for using Oracle 12 and 19 databases with SSL/TLS connection](#) on page 14.

4. Start new web servers and point them to the new database engine.
5. After you restart Jive, the upgrade process begins, and the database schema is upgraded.

Note that if you upgrade from Jive 7.x.x, on Oracle databases, you may get an error because the null constraint already exists. The error may look like this:

```
java.sql.SQLException: ORA-01442: column to be modified to NOT NULL is already NOT NULL
```

This error does not affect the installation, and you should ignore it.

## Postgres database best practices

For best results, configure and maintain your Postgres database according to the vendor's requirements and use the Jive best practices.

- Configure the PostgreSQL `ph_hba.conf` to allow md5 communication between the application servers and the databases. This setting should be applied to all instances.
- Create users for each of the databases. Typically, the user names are identified as the DB names.
- Create the databases using the `OWNER` option using the applicable role created previously.
- For performance and troubleshooting, set the `logging_collector` to `on`, and ensure that Postgres logs are time-stamped. Use caution in busy environments with log settings, because excessive logging can generate critical load on the server.
- Set the values that may impact performance. The PostgreSQL `postgresql.conf` file is, by default, set to run using extremely limited resources. Settings where default values can impact performance include but are not limited to, the following:
  - `shared_buffers`
  - `checkpoint_segments`
  - `checkpoint_completion_target`
  - `work_mem`
- Configure monitoring on the system to assess the following metrics, at a minimum:
  - DB Size
  - Available Memory
  - Connections
  - Read/Write Activity
  - Transaction rates

## Oracle database best practices

For best results, you should configure and maintain your Oracle database according to the vendor's requirements and use the Jive best practices.

- If you're setting up new systems, the Core and Activity Engine databases can be set up as typical OLTP databases. Set up the Analytics database using the OLAP template.
- Oracle JDBC Drivers must be installed on the application services before the Jive installation. For more information, see [Oracle database drivers and configurations](#) on page 13.
- To ensure good performance, set the following JVM property on each of the web application nodes:

```
-Doracle.jdbc.maxCachedBufferSize=12
```

- Ensure that the target database allows the Jive application's Setup wizard access to create tables. After you install the Jive application, you finish the installation (or upgrade) by using Jive's Setup wizard to connect to the target database. The Setup wizard creates or upgrades the tables it needs. So, you need to ensure that permission to create tables is granted, at least until the upgrade or installation is completed and the application has been restarted.

For example, if you are using an Oracle database for a new installation, you'll be creating a user representing a schema. That schema will be empty until the Setup wizard creates tables in it. Here's an example of a script for creating such a user:

```
CREATE USER coredb IDENTIFIED BY changeme DEFAULT TABLESPACE jivedata;  
GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE SEQUENCE, CREATE  
PROCEDURE, CREATE SYNONYM, CREATE TYPE to coredb;  
ALTER USER coredb QUOTA UNLIMITED ON jivedata;
```

The username can be any legal Oracle identifier – `coredb` is just an example. The application does not require a dedicated tablespace, but many DBAs have that practice. The tablespace must be created separately. For more information about the `CREATE USER` statement, refer to the Oracle documentation.

---

### Caution:

The database user must not be granted view access to any schema other than the one it owns. During upgrades, the application reads metadata about its tables defined in the database. Since the application does not know what the default schema is, it may inadvertently retrieve information about its tables defined in other schemas. If you have set up application instances by using other schemas in the same database, this can result in erroneous information being passed to the application.

If you absolutely must use a database user with access to other schemas, ensure that there is only one set of application tables defined across all schemas to prevent this object name conflict.

---

# Oracle database drivers and configurations

In addition to the following recommendations, be sure and follow vendor documentation for driver installation best practices.

Component	Supported driver version	Notes
Web application	11.2.0.4 (ojdbc6.jar) 12.2.0.1 (ojdbc8.jar)	Type the following from a command line as the jive user: <pre>jive set webapp.custom_ld_library_path_additions   path/to/driver jive set webapp.custom_classpath_additions   path/to/your_JDBC_Driver.jar</pre> where <i>path/to/driver</i> is the path to the Oracle JDBC driver and <i>your_JDBC_Driver.jar</i> is the name of the JAR file.
Activity Engine	11.2.0.4 (ojdbc6.jar) 12.2.0.1 (ojdbc8.jar)	Type the following from a command line as the jive user: <pre>jive set eae.custom_ld_library_path_additions   path/to/driver jive set eae.custom_classpath_additions   path/to/your_JDBC_Driver.jar</pre> where <i>path/to/driver</i> is the path to the Oracle JDBC driver and <i>your_JDBC_Driver.jar</i> is the name of the JAR file.

## Additional configuration for Oracle 11gR2, 12, 19

Customers using Oracle 11gR2, 12, 19 need to add a `fetchSizeSupported` entry in the `database` element of the `jive_startup.xml` file as shown:

```
<jive>
  <setup>true</setup>
  <temp/>
  <locale>
    <characterEncoding>UTF-8</characterEncoding>
  </locale>
  <database>
    <defaultProvider>
      <password encrypted="true"/>
      <connectionTimeout>60</connectionTimeout>
      <maxConnections>50</maxConnections>
      <username>username</username>
      <driver>oracle.jdbc.driver.OracleDriver</driver>
      <minConnections>25</minConnections>
      <serverURL>jdbc:oracle:oci:@host:1521/db</serverURL>
    </defaultProvider>
    <fetchSizeSupported>>false</fetchSizeSupported>
  </database>
  <connectionProvider>
    <className>com.jivesoftware.base.database.DefaultConnectionProvider</className>
  </connectionProvider>
</jive>
```

## Required libraries for RHEL7

Note that distributions based on RHEL7 do not always install `libaio.so` by default. This library is required to use the Oracle driver. On app servers running Red Hat 7 or CentOS 7 that use Oracle as a database, make sure to run the following command:

```
# apt-get update && apt-get install libaio1</codeblock>
```

## Additional configuration for using Oracle 12 and 19 databases with SSL/TLS connection

It is possible to set up the connectivity between the application and database servers to encrypt the data in transit, leveraging Oracle's support for SSL/TLS connections. Here you can find setup instructions.

Even though the database you are using can run in Cloud on AWS RDS or on-premise, the setup includes the same basic steps:

1. Enabling TLS encryption support.
2. Setting up application servers.
3. Installing Jive

### Prerequisites and limitations

In this example, we set up Jive on Oracle 19.0 database hosted by AWS RDS using TLS v1.0 and cypher `SSL_RSA_WITH_AES_256_CBC_SHA`. This setup has been tested on Oracle 12.1 and 19.0 and is applicable for 12.x and 19.x databases. Implementation details may vary depending on the actual DB hosting and version.

The instructions apply to AWS RDS and on-premise installations. Any notable distinctions are noted specifically.

If you are using Oracle databases on AWS RDS, peruse relevant sections of the Amazon Relational Database Service (RDS) documentation on AWS Portal:

- [Using SSL with an Oracle DB instance](#) for the necessary Option Group parameters for your database version.
- [Oracle Secure Sockets Layer](#) for the details on which TLS protocols are supported by your database version.
- [Cipher suites for the Oracle SSL option](#) for cypher suites that can be associated with your database version.

### Enabling SSL/TLS encryption support in databases

#### On AWS RDS

If you are running the Oracle database on AWS RDS, the option group configured for the database must include the SSL option. You can find the necessary parameters for your database version in [Using SSL with an Oracle DB instance](#).

In this example:

1. To ensure TLS encryption support, associate an options group that includes an SSL option with your Amazon RDS Oracle DB. For details, refer to [Using SSL with an Oracle DB instance](#).
2. Ensure network connectivity is in place for the TLS port. By default, it is port 2484 on AWS RDS:

Options				
Name	Persistent	Permanent	Port	Security Groups
SSL	No	No	2484	Security Group sg-0f6fb9fad23fe727c

### On-premise

If you are running Oracle on-premise, contact your database administrator to have the server set up to support TLS encryption.

### Configuring application servers

Follow these steps to configure application servers for Jive:

1. Install the Instant Client and Identity and Access Management tool for Oracle certificate wallet configuration. They can be downloaded from the Oracle portal:

- [Oracle Instant Client Downloads](#)
- [Identity & Access Management Downloads](#)

For more information, refer to the Oracle documentation.

2. Configure SQL\*Plus to use SSL with an Oracle DB instance.

Refer to instructions in [Configuring SQL\\*Plus to use SSL with an Oracle DB instance](#) in the [Amazon Relational Database Service \(RDS\) documentation](#) on the AWS portal. For more information, see [Setting Up Oracle Wallet Using ORAPKI](#) on the Oracle portal.

3. Validate the connectivity from the application servers using SQL\*Plus by using the following command:

```
sqlplus '<mydbuser>@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCPS) (HOST = <endpoint>) (PORT = <ssl port number>))(CONNECT_DATA = (SID = <database name>)))'
```

For example:

```
sqlplus 'jiveadmin@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCPS)(HOST =
jive-db.us-east-1.rds.amazonaws.com) (PORT = 2484)) (CONNECT_DATA = (SID = ORCL)))'
```

For more information, see [Connecting to an Oracle DB instance using SSL](#).

4. Import the SSL Certificate for the appropriate Certificate Store into Jive's JVM local Certificate Store by using the following command:

```
sudo su - jive
java/bin/keytool -importcert -file <path to certificate or pem file> -keystore
java/jre/lib/security/cacerts -storetype BCFKS -providername BCFIPS -alias
"CustomOracleCertificate" -storepass changeit
```

---

**Note:** AWS RDS Certificate is already included in Jive 9.5 onward, so **this step may be unnecessary** if the Certificate Authority being used is a well-known public entity, such as the one used by AWS.

---

For more information, refer to Steps 1–4 in [Connecting to an Oracle DB instance using SSL](#) in the [Amazon Relational Database Service \(RDS\) documentation](#) on the AWS portal.

5. On the **Webapp nodes**, add references for the OJDBC driver installed by Oracle Instant Client by using the following command:

```
jive set webapp.custom_ld_library_path_additions
/usr/local/jive/lib/instantclient_12_2
jive set webapp.custom_classpath_additions
/usr/local/jive/lib/instantclient_12_2/ojdbc8.jar
```

6. On the **EAE server**, add references for the OJDBC driver installed by Oracle Instant Client by using the following command:

```
jive set eae.custom_ld_library_path_additions
/usr/local/jive/lib/instantclient_12_2
jive set eae.custom_classpath_additions
/usr/local/jive/lib/instantclient_12_2/ojdbc8.jar
```

## Configuring Jive for TLS/SSL usage

After configuring servers, install Jive setting up the TLS/SSL connection to the databases and then configuring the Analytics database as follows:

1. Run Jive installer on all nodes by using the following command:

```
jive start
```

2. Install Jive normally and set up the connection strings specifying the secure connection properties on the **Database Setting** step for the Core database and on the **Activity Engine Settings** step for the EAE database.

For example:

**Figure 1: An example of Core Database settings**

**Figure 2: An example of Activity Engine (EAE) settings**

Jive installs normally and you should be able to use TLS/SSL for the instance.

- For the Analytics database, specify the secure protocol and port in addition to the usual configuration. For details, refer to [Configure Analytics Database](#) in the Jive Analytics Help.

## Microsoft SQL server database best practices

For best results, you should configure and maintain your Microsoft SQL database according to the vendor's requirements and use the Jive best practices.

### Installation

- SQL Server installations may be standalone instances or clustered. Mirrors with automatic-failover are not supported, but you can use mirrors for redundancy or DR purposes.
- The core application and the Activity Engine run using the jTDS MSSQL Driver version 1.3.x or higher. The Analytics database is not supported on MSSQL.
- Set your classpath to include the jTDS Driver by running the following from a command line as the jive user:

```
jive set webapp.custom_classpath_additions path/JDBC_Driver.jar
```

where `path` is the path to the jTDS JDBC driver and `JDBC_Driver.jar` is the name of the JAR file.

---

### Note:

If you have implemented SSL to secure traffic between the web application nodes and the database, an issue with the jTDS driver can prevent the Jive application from starting. To work around this problem, run the following commands on each web application node:

```
jive set webapp.custom_jvm_args " -Djsse.enableCBCProtection=false"
jive set eae.custom_jvm_args " -Djsse.enableCBCProtection=false"
```

---

### Case sensitivity

SQL Server instance default collation is case-insensitive and accent-sensitive. If database collation is not specified, the instance collation is used. **Latin1\_General\_CI\_AS** is recommended for typical installations. In special cases where the Jive platform is configured to allow case-sensitive user names (for example, where `jsmith` and `JSmith` are different logins), **Latin1\_General\_CS\_AI** is required so that case sensitivity may be utilized in indexing and string searches.

You can find more information on adjusting SQL Server settings in the SQL Server documentation: Setting and Changing the Server Collation Setting ([http://msdn.microsoft.com/en-us/library/ms179254\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms179254(SQL.90).aspx)) and Changing the Database Collation ([http://msdn.microsoft.com/en-us/library/ms175835\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms175835(SQL.90).aspx)).

## MySQL database best practices

For best results, configure and maintain your MySQL database according to the vendor's requirements and use the Jive best practices.

### UTF-8 requirement

Jive requires the UTF-8 character set. Use the following steps to ensure it's correctly set up:

1. Use the following command to set the database default character set to UTF-8:

```
ALTER DATABASE jivecoredb DEFAULT CHARACTER SET utf8;
```

2. After installation, add the following to the `<database>` section of the `jive_start-up.xml` file:

```
<mysql><useUnicode>true</useUnicode></mysql>
```

3. In the Jive Admin Console, under **System > Management > Locale**, ensure that the character set setting is set to UTF-8.

## Installing the driver

The core application and the Activity Engine run using the MySQL JDBC Connector/J Driver. The Analytics database is not supported by MySQL.

You can set your classpath to include the MySQL JDBC Connector/J Driver by running the following from a command line as the jive user:

```
jive set webapp.custom_classpath_additions path/JDBC_Driver.jar
```

where `path` is the path to the MySQL JDBC driver and `JDBC_Driver.jar` is the name of the JAR file.

## Other best practices with MySQL

Default configuration can vary significantly by version, so it is important to consider the following:

- The default storage engine and all jive tables should use the InnoDB storage engine. Earlier versions of MySQL used BDB\MyISAM as the default storage engine.
- Ensure that the InnoDB Buffer is adequately sized. For dedicated installations with InnoDB storage engines, the buffer is typically set to 80% of the total server available memory.
- Storing attachments in your Jive database is not recommended. If you must do this, ensure that the maximum attachment size is allowed in the MySQL configuration. For more information, see Packet Too Large in the MySQL 5.1 Reference Manual at <http://dev.mysql.com/doc/refman/5.1/en/packet-too-large.html>.
- With case sensitivity, to avoid upgrade problems, use lower-case table names in MySQL and set the `lower_case_table_names` system variable to 1 in the MySQL configuration. For more information, see Server System Variables in the MySQL 5.0 Reference Manual at [http://dev.mysql.com/doc/refman/5.0/en/server-system-variables.html#sysvar\\_lower\\_case\\_table\\_names](http://dev.mysql.com/doc/refman/5.0/en/server-system-variables.html#sysvar_lower_case_table_names).

## Quick database setup for evaluations

You can use the quick database setup procedure for evaluations and other light installations.

You can use the following steps to install PostgreSQL and create the databases you need to evaluate Jive. You need to create Core, Activity Engine, and Analytics databases. Use the full database instructions to install your production databases, as described in the [Database configuration and best practices](#) on page 6 section.

1. Browse Yum.postgres.org (<http://yum.postgresql.org/repopackages.php>) to find a PostgreSQL 9.0 or higher package to use for your Jive evaluation.
2. Download the appropriate rpm for your operating system to configure the proper yum repository.

```
For example, type wget http://yum.postgresql.org/9.1/redhat/rhel-5-x86_64/pgdg-centos91-9.1-4.noarch.rpm
```

3. Install the yum repository through RPM.

For example, type `rpm -ivh pgdg-centos91-9.1-4.noarch.rpm`

#### 4. Install PostgreSQL.

For example, type `yum -y install postgresql91-server`

#### 5. Initialize the database.

For example, type `/etc/init.d/postgresql-9.1 initdb`

#### 6. Change the `pg_hba.conf`.

For example, type:

```
cat <<EOF > /var/lib/pgsql/9.1/data/pg_hba.conf
local all all ident
host all all <application server IP address>/32 md5
EOF
```

#### 7. Start the server.

For example, type `/etc/init.d/postgresql-9.1 start`

#### 8. Set up the databases and users.

For example, type:

```
cat <<EOF | su - postgres -c psql
create user core with password 'core';
create database core owner core encoding 'UTF-8';
create user eae with password 'eae';
create database eae owner eae encoding 'UTF-8';
create user analytics with password 'analytics';
create database analytics owner analytics encoding 'UTF-8';
EOF
```