



CONFLUENCE

DOCUMENTATION

0.1 Application Server Configuration	2
0.0.1 Application Server URL encoding	2
0.0.1.1 Configuring Tomcat's URI encoding	2
0.0.2 Managing Application Server Memory Settings	3
0.0.3 Switching to Apache Tomcat	3
0.1 Database Configuration	6
0.0.1 Known Issues For Supported Databases	7
0.0.1.1 Configuring Database Character Encoding	7
0.0.2 Database Setup Guides	10
0.0.2.1 Database JDBC drivers	10
0.0.2.2 Database Setup for Oracle	10
0.0.2.3 Database Setup for SQL Server	13
0.0.2.3.1 Configuring a SQL Server Datasource in Apache Tomcat	14
0.0.2.4 Database Setup For Any External Database	16
0.0.2.5 Database Setup for PostgreSQL	17
0.0.2.5.1 Configuring a PostgreSQL Datasource in Apache Tomcat	21
0.0.2.6 Database Setup For MySQL	23
0.0.2.6.1 Configuring a MySQL Datasource in Apache Tomcat	27
0.0.3 Creating Database Schema Manually	28
0.0.4 Migrate to Another Database	29
0.0.4.1 The Embedded HSQLDB Database	31
0.0.5 Troubleshooting the Embedded Database (HSQL DB)	31
0.0.5.1 Connecting to HSQLDB using DBVisualizer	32
0.0.5.1.1 Database Tables Reference	35
0.0.6 Troubleshooting External Database Connections	35
0.0.6.1 Configuring database query timeout	36
0.0.7 Improving Database Performance	36
0.0.7.1 Creating a Lowercase Page Title Index	37
0.1 Webserver Configuration	39
0.0.1 Apache and Apache Connector Tips	39
0.0.2 Configure Web Proxy Support for Confluence	39
0.0.3 Running Confluence behind Apache	41
0.0.3.1 General Apache Configuration Notes	42
0.0.3.2 Using Apache with mod_jk	42
0.0.3.3 Using Apache with mod_proxy	44
0.0.3.4 Using Apache with virtual hosts and mod_proxy	48
0.0.3.5 Using mod_rewrite to Modify Confluence URLs	49
0.1 Start Confluence Automatically on System Startup	50
0.0.1 Start Confluence Automatically on Linux	50
0.0.2 Start Confluence Automatically on Windows as a Service	53
0.1 Confluence Data Model	55
0.1 Known Issues with Enterprise or Webhosting environments	65
0.1 Setting Up Public Access	66
0.1 Setting Up a Mail Session for the Confluence Distribution	66
0.1 Troubleshooting SQL Exceptions	67

Confluence Configuration Guide

The pages listed below contain instructions on configuring Confluence. If you cannot find what you are looking for, try the search box in the left-hand navigation panel.

- [Application Server Configuration](#)
 - [Application Server URL encoding](#)
 - [Managing Application Server Memory Settings](#)
 - [Switching to Apache Tomcat](#)
- [Database Configuration](#)
 - [Known Issues For Supported Databases](#)
 - [Database Setup Guides](#)
 - [Creating Database Schema Manually](#)
 - [Migrate to Another Database](#)
 - [Troubleshooting the Embedded Database \(HSQL DB\)](#)
 - [Troubleshooting External Database Connections](#)
 - [Improving Database Performance](#)
- [Webserver Configuration](#)
 - [Apache and Apache Connector Tips](#)
 - [Configure Web Proxy Support for Confluence](#)
 - [Running Confluence behind Apache](#)
- [Start Confluence Automatically on System Startup](#)
 - [Start Confluence Automatically on Linux](#)
 - [Start Confluence Automatically on Windows as a Service](#)
- [Confluence Data Model](#)
- [Known Issues with Enterprise or Webhosting environments](#)
- [Setting Up Public Access](#)
- [Setting Up a Mail Session for the Confluence Distribution](#)
- [Troubleshooting SQL Exceptions](#)

RELATED CONTENT

[Tracking Customisations Made to your Confluence Installation](#)
[Documentation Home](#)
[Confluence Administrator's Guide](#)
[Frequently Asked Questions](#)
[Confluence Community](#)
[Configuring Confluence](#)
[Development Hub](#)
[Plugins and Extensions](#)

Application Server Configuration

The following pages contain information about configuring your application server for Confluence:

- [Application Server URL encoding](#)
- [Managing Application Server Memory Settings](#)
- [Switching to Apache Tomcat](#)

Application Server URL encoding

Application servers may have different settings for character encodings. We strongly suggest setting this to **UTF-8** where possible.

Information on setting the character encoding is available at:

- [Configuring Tomcat's URI Encoding](#)

Configuring Tomcat's URI encoding

By default, Tomcat uses ISO-8859-1 character encoding when decoding URLs received from a browser. This can cause problems when Confluence's encoding is UTF-8, and you are using international characters in attachment or page names.

1. Edit `conf/server.xml` and find the line where the Coyote HTTP Connector is defined. It will look something like this, possibly with more parameters:

```
<Connector port="8090" />
```

2. Add a `URIEncoding="UTF-8"` property to the connector:

```
<Connector port="8090" URIEncoding="UTF-8" />
```

3. Restart Tomcat

If you are using mod_jk

You should apply the same URIEncoding parameter as above to the AJP connector if you are using mod_jk, and add the following option to your Apache mod_jk configuration:

```
<Connector port="8009" protocol="AJP/1.3" URIEncoding="UTF-8" />
```

```
JkOptions +ForwardURISCompatUnparsed
```

More information using Apache with Tomcat

For comprehensive examples of how to use Tomcat and Apache with Confluence, see [Running Confluence behind Apache](#).

Managing Application Server Memory Settings

The minimum and maximum JVM heap space allocated to the application server affects performance. Confluence administrators may wish to modify this value from the defaults [depending on their server load](#). This document only provides guidelines rather than rules, so administrators optimising for performance should use this document as a starting point only.



For a comprehensive overview of memory management, and memory tuning in Confluence under Sun JRE, please read [Garbage Collector Performance Issues](#)

Testing For Optimum Memory Settings

In the general case, both JIRA & Confluence users will benefit from setting the minimum and maximum values identical. In larger installations, there is benefit to memory tuning, if there is a perceived performance issue. If you are experiencing Out of Memory Heap errors, try doubling the -Xmx and -Xms values for your installation to see if this resolves or helps resolve your issue. If not, please lodge a support ticket as there may be other factors contributing.

Memory usage is most likely to be maximised under peak load, and when [creating a site XML backup](#). In many cases, the backup can be the cause of the OOM, so increase -Xmx values and verify if a backup was occurring at the time of OOM. A quick rule of thumb for gauging the success of a memory adjustment is using simple anecdotal evidence from users. Is it snappier? The same? How does it handle while a backup is occurring?



Atlassian recommends in normal use, to disable the XML backup and use a [Production Backup Strategy](#).

- If you normally perform manual XML site backups on your server, test your maximum memory requirements by performing a site XML backup while the server is under maximum load
- If you do not create manual XML site backups, simply monitor the server while under maximum load

Applying Memory Settings

See [Fix Out of Memory Errors by Increasing Available Memory](#).

Related Topics

- [Garbage Collector Performance Issues](#)
- [Fix Out of Memory Errors by Increasing Available Memory](#)
- [Server Hardware Requirements Guide](#)
- [Performance Tuning](#)
- [Troubleshooting Slow Performance Using Page Request Profiling](#)
- [Tomcat JVM options and Modify the Default JVM Settings](#)

Switching to Apache Tomcat

Apache Tomcat is the only application server supported for Confluence. To move Confluence from an application server (e.g. WebSphere) to Tomcat using the same database, follow the instructions below.

Please note, you cannot simply copy the WAR file or expanded WAR directory from an old Confluence EAR/WAR version in the old application server to Tomcat. **This will not work.**

Follow these instructions:



- 1. Before You Start
- 2. Backing Up
- 3. Switching Application Servers
- 4. Applying Customisations
 - Confluence Server
 - Plugins
 - Look and Feel
 - Performance
 - Advanced Customisations
- 5. Testing Confluence

1. Before You Start

1. The following instructions will only work if you are running the same major version of Confluence on both application servers. If you are running different major versions of Confluence, you will need to [upgrade Confluence](#) before you can switch to Tomcat.
2. Note that you need current [software maintenance](#), as the process for changing application servers involves installing Confluence or Confluence EAR-WAR.
3. If the environment (e.g. the database system, the operating system and so on) that you are running Confluence in has changed, please ensure it still complies with the Confluence [System Requirements](#).
4. If you are using an external database, familiarise yourself with all [known issues](#) for your specific database. Also make sure the Confluence database connector principal (the database user login) has sufficient permissions to modify the database schema.
5. Note any customisations that you have made to Confluence, e.g. enabled/installed plugins, modified layouts, custom themes, etc. You will need to reapply these after you have switched to Tomcat. You can view the list of customisations in the [Reapplying Customisations](#) section below.
6. We recommend that you **do not** run any other applications in your Tomcat application server that is running Confluence, to prevent performance issues.

2. Backing Up

Before you switching to Tomcat, you must back up the following:

1. **Back up your Confluence Home directory.** The Confluence Home directory is the folder where Confluence stores its configuration information, search indexes and page attachments. If you are using the embedded HSQLDB database supplied for evaluation purposes, the database files are also stored in this directory.
 -  **Tip:** Another term for 'Home directory' would be 'data directory'. The location of the Home directory is stored in a configuration file called `confluence-init.properties`, which is located inside the `confluence/WEB-INF/classes` directory in your Confluence Installation directory.
2. **Back up your database.** Perform a manual backup of your external database before proceeding with the upgrade and check that the backup was created properly. If you are not a database expert or unfamiliar with the backup-restore facilities of your database, you should try to restore the backup to a different system to ensure that the backup worked before proceeding. This recommendation is not specific to Confluence usage, but it is good practice to ensure that your database backup is not broken.
 -  The 'embedded database' is the HSQLDB database supplied with Confluence for evaluation purposes, you don't need to back it up since it is stored in the home directory. But you should not use this database for production systems anyway, so if you happen to accidentally still use HSQLDB in a production system, please [migrate to a proper database](#) **before** the upgrade.
3. **Back up your Confluence Installation directory** (if you are using Confluence) **or your Confluence webapp** (if you are using Confluence EAR-WAR edition). The 'Confluence Installation directory' is the directory into which the Confluence application files and libraries have been unpacked (unzipped) when Confluence was installed. Confluence does not modify or store any data in this directory. This directory is also sometimes called the 'Confluence Install directory'.

3. Switching Application Servers

1. Install Confluence on your new application server. We recommend that you [install Confluence](#) (from the zip file) as it is preconfigured with Tomcat. If you want more control over the installation process, you can [install Confluence EAR-WAR on Tomcat](#) however this requires more manual configuration.

Regardless of which method you choose, as part of the installation process:

- If you are connecting to your database via a standard JDBC connection, enter the URL, username and password for your existing database.

- If you are connecting to your database via datasource, use the settings for your existing database when you configure the JDBC datasource in your new server. Refer to the appropriate guide below:
 - [Configuring a MySQL Datasource in Apache Tomcat](#)
 - [Configuring a SQL Server Datasource in Apache Tomcat](#)
 - [Configuring a PostgreSQL Datasource in Apache Tomcat](#)
- 2. Copy the following files from your old Confluence installation to your new one:
 - {CONFLUENCE_INSTALL}\confluence\WEB-INF\classes\confluence-init.properties
 - {CONFLUENCE_INSTALL}\confluence\WEB-INF\classes\atlassian-user.xml
 - {CONFLUENCE_INSTALL}\confluence\WEB-INF\classes\osuser.xml (*copy this over if you are using JIRA user management*)
 - {CONFLUENCE_INSTALL}\confluence\WEB-INF\classes\seraph-config.xml (*copy this over if you using custom SSO*)
 - {CONFLUENCE_INSTALL}\confluence\WEB-INF\web.xml (*copy this over if you have previously modified it, e.g. to configure a datasource*)
- 3. Make sure you shutdown the old server before you startup the new one.
- 4. If you are running the new application server on a different machine to the old one, carry out the following actions as soon as you start the new server:
 - [Re-index your data.](#)
 - Make sure that the [attachments location](#) is valid for the new server.
- 5. If you have applied special settings to their Confluence server and/or Confluence look and feel, you will need to reapply these customisations as described in [below](#).

4. Applying Customisations

After switching to Tomcat, you need to review any customisations and other special configurations you previously used for your Confluence instance, and re-apply if necessary. This section also contains some Tomcat-specific customisations that you may wish to considering applying, if you haven't used Confluence with Tomcat before.



Before you apply customisations

Please ensure that your Confluence installation works correctly on Tomcat without any customisations before you apply any of customisations listed below. This will make it easier to identify problems, if you run into trouble during the switch to Tomcat.

Confluence Server

- For long-term use, we recommend that you configure Confluence to start automatically when the operating system restarts. For Windows servers, this means [configuring Confluence to run as a Windows service](#).
- If you are using the Confluence edition and you have previously defined a **CATALINA_HOME** environment variable, please check that it points to the correct path for the new Confluence Tomcat server.
- If you were previously running **Confluence on a non-standard port**, edit your new <Installation-Directory>\conf\server.xml file as described in [Change listen port for Confluence](#).

Plugins

- If you were previously using any **plugins**, install the latest compatible version and disable any plugins that are incompatible with you new instance of Confluence. The easiest way to do this is to use the [Plugin Repository](#) in the Confluence Administration Console.

Look and Feel

- If you are using any **customised themes**, please check that they are displaying as expected. Some further customisation may be required to ensure compatibility with your new version of Confluence.
- If you had previously customised the **default site or space layouts**, you will need to reapply your changes to the new defaults as [described here](#). Please do not just copy your VM (velocity) files across. Ensure that Confluence works without your custom layouts then apply the layout via the Confluence Administration console.

Performance

- If the load on your Confluence instance is high, you may need more simultaneous connections to the database. Read more about this in the [Performance Tuning](#) guide.
- If you had previously modified the **memory flags** (Xms and Xmx) in either the <Installation-Directory>\bin\setenv.sh or the <Installation-Directory>\bin\setenv.bat file, you may want to make the modifications in your new installation. The parameters are specified in the JAVA_OPTS variable. See [Fix Out of Memory Errors by Increasing Available Memory](#) for more information.

Advanced Customisations

- If you were previously running **Confluence over SSL**, you will need to reapply your configuration as described in [Running](#)

[Confluence Over SSL or HTTPS.](#)

- If you were using a custom SSO authenticator, change seraph-config.xml to the correct authenticator.
- If you had changed the [Confluence interface text](#), you will need to copy over the ConfluenceActionSupport.properties file.
- If you had previously modified the Confluence **source code**, you will need to reapply your changes to the new version.

5. Testing Confluence

Make sure you **test Confluence on the new server** before deploying it in production.

The [Working with Confluence Logs](#) document contains the locations for the application logs, if you need to refer to them.

Database Configuration

This document provides information on configuring an external database.

The Embedded Database for Evaluation Purposes

The Confluence installation includes an embedded [HSQLDB](#) database, supplied for the purpose of **evaluating Confluence**.

If you are using the embedded database, the database files are stored in the \database folder under your [Confluence Home directory](#). See also [Important Directories and Files](#).



Embedded Database is Not Suitable for Production Instances of Confluence

Production instances of Confluence should use an [external database](#). When using the default HSQLDB database, you run the risk of unrecoverable data loss due to not being [transaction safe](#).

- Corruption is occasionally encountered after sudden power loss and can usually be corrected using [this data recovery procedure](#).
- HSQLDB is still suitable for evaluation purposes, but the risk can only be eliminated by switching databases. External databases may also provide superior speed and scalability.

Selecting an External Database

The XML backup built into Confluence is not well suited for database migration for large data sets (see [Production Backup Strategy](#)). Choose your database wisely. If you need to migrate later, Atlassian support will refer you to a third party database migration tool. Vote for [CONF-12599](#) to add a more robust strategy for large implementation migrations.

Below is more information on selecting and migrating to an external database:

- [Migrating to a Different Database](#)
- [List Of Supported Databases](#)
- [Known Issues For Supported Databases](#)

Database Setup

Setup instructions are shown below.

- [Database JDBC drivers](#)
- [Database Setup for Oracle](#)
- [Database Setup for SQL Server](#)
- [Database Setup For Any External Database](#)
- [Database Setup for PostgreSQL](#)
- [Database Setup For MySQL](#)

Optimising Database Performance

To improve database responsiveness:

- [Improving Database Performance](#)
- [Known Issues For Supported Databases](#)

Database Troubleshooting

For solving database-related problems:

- [Troubleshooting External Database Connections](#)
- [Troubleshooting the Embedded Database \(HSQL DB\)](#)
- [Interpreting DB2 error codes](#)
- [Known Issues For Supported Databases](#)

Obtain technical support from [Troubleshooting Problems](#) and [Requesting Technical Support](#).

Known Issues For Supported Databases

Supported Databases

Please refer to the [Supported Platforms](#) topic.

Troubleshooting

[Known Issues For PostgreSQL](#)

[Known Issues For Oracle](#)

[Known Issues For MySQL](#)

[Known Issues For SQL Server](#)

[Known Issues For Db2](#)

[Configuring Database Character Encoding](#)

Configuring Database Character Encoding

The database used with Confluence should be configured to use the same character encoding as Confluence. **The recommended encoding is Unicode UTF-8.**

There are two places where character encoding may need to be configured:

- when creating the database
- when connecting to the database (JDBC connection URL or properties).

The configuration details for each type of database are different. Some examples are below.

On this page:

- [JDBC connection settings](#)
- [Creating a UTF-8 database](#)
- [Updating existing database to UTF-8](#)

 *The information on this page does not apply to **Confluence OnDemand**.*

JDBC connection settings

MySQL

Append "useUnicode=true to your JDBC URL:

```
jdbc:mysql://hostname:port/database?useUnicode=true&characterEncoding=utf8
```




If you are modifying `confluence.cfg.xml` directly rather than via the Confluence Installation GUI, you'll need to escape our XML token and will break the syntax when the XML is parsed. An effective URL could be similar to:

```
<property
  name="hibernate.connection.url">jdbc:mysql://hostname:port/database?useUnicode=true&amp;ch
```

Creating a UTF-8 database

MySQL

1. Create a UTF-8 database with binary UTF-8 collation.

 Binary UTF-8 provides case-sensitive collation.

```
CREATE DATABASE confluence CHARACTER SET utf8 COLLATE utf8_bin;
```

2. You will also need to set the Server Characterset to utf8. This can be done by adding the following in `my.ini` for Windows or `my.cnf` for other OS. It has to be declared in the Server section, which is the section after `[mysqld]`:

```
[mysqld]
default-character-set=utf8
```

3. Use the `status` command to verify database character encoding information.

Screenshot: Using the Status Command to Verify Database Character Encoding

```
mysql> CREATE DATABASE confluence CHARACTER SET utf8 COLLATE utf8_bin;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| confluence         |
| confluencedb       |
| mysql              |
| test               |
+-----+
5 rows in set (0.02 sec)

mysql> use confluence;
Database changed
mysql> status;
-----
C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql.exe  Ver 14.12 Distrib 5.0.83,
for Win32 (ia32)

Connection id:          1800
Current database:       confluence
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.0.83-community-nt MySQL Community Edition (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:    utf8
Db characterset:        utf8
Client characterset:    utf8
Conn. characterset:     utf8
TCP port:               3306
Uptime:                 20 hours 56 min 23 sec
```

4. In some cases, the individual tables collation and character encoding may differ from the one that the database as a whole has been configured to use. Please use the command below to ensure all tables within your Confluence database are correctly configured to use UTF-8 character encoding and binary UTF-8 collation:

```
use confluence;
show table status;
```

Check for the value listed under the **Collation** column, to ensure it has been set to `utf8_bin` (that is, case-sensitive) collation for all tables.

If not, then this can be changed by the following command, executed for each table in the Confluence database:

```
ALTER TABLE tablename CONVERT TO CHARACTER SET utf8 COLLATE utf8_bin;
```

Please substitute the `<tablename>` above, with each table within the confluence database.

Relevant MySQL manual for more detailed explanation:

- [Specifying Character Sets and Collations documentation.](#)
- [Connection Character Sets and Collations.](#)
- [SHOW TABLE STATUS Syntax.](#)
- [ALTER TABLE Syntax.](#)

PostgreSQL

```
CREATE DATABASE confluence WITH ENCODING 'UNICODE';
```

Or from the command-line:

```
$ createdb -E UNICODE confluence
```

For more information see the [PostgreSQL documentation](#).

For PostgreSQL running under Windows

Please note that international characters sets are only fully supported and functional when using PostgreSQL 8.1 and above under Microsoft Windows.

For PostgreSQL running under Linux



Please make sure you check the following to ensure proper handling of international characters in your database

When PostgreSQL creates an initial database cluster, it sets certain important configuration options based on the host environment. The command responsible for creating the PostgreSQL environment `initdb` will check environment variables such as `LC_CTYPE` and `LC_COLLATE` (or the more general `LC_ALL`) for settings to use as database defaults related to international string handling. As such it is important to make sure that your PostgreSQL environment is configured correctly **before you install Confluence**.

To do this, connect to your PostgreSQL instance using `psql` and issue the following command:

```
SHOW LC_CTYPE;
```

If `LC_CTYPE` is set to either "C" or "POSIX" then certain string functions such as converting to and from upper and lower case will not work correctly with international characters. Correct settings for this value take the form `<LOCALE> . <ENCODING>` (`en_AU.UTF8` for example).

If your `LC_CTYPE` is incorrect please check the [PostgreSQL documentation for information on configuring database localisation](#). It is not easy to change these settings with a database that already contains data.

Updating existing database to UTF-8

MySQL database with existing data



For an existing database

If you're using a existing database, confirm the Character Encoding by executing the query:
`SHOW VARIABLES LIKE 'character%';` and `SHOW VARIABLES LIKE 'collation%';`.
 The results should be UTF-8.



Before proceeding with the following changes, please **backup your database**.

This example shows how to change your database from latin1 to utf8.

1. Dump the database to a text file using `mysqldump` tool from the command-line :
`mysqldump -p --default_character-set=latin1 -u <username> --skip-set-charset confluence > confluence_database.sql`
2. copy `confluence_database.sql` to `confluence_utf8.sql`
3. Open `confluence_utf8.sql` in a text editor and change all character sets from 'latin1' to 'utf8'
4. Encode all the latin1 characters as UTF-8:
`recode latin1..utf8 confluence_utf8.sql` (the recode utility is described at <http://directory.fsf.org/recode.html>; it can actually be downloaded from <http://recode.progiciels-bpi.ca/>, and is available for Ubuntu via `apt-get`)

In MySQL:

1. DROP DATABASE confluence;
2. CREATE DATABASE confluence CHARACTER SET utf8 COLLATE utf8_bin;

Finally, reimport the UTF-8 text file:

1. mysql -u <username> -p --default-character-set=utf8 --max_allowed_packet=64M confluence < /home/confluence/confluence_utf8.sql

To support large imports, the parameter '--max_allowed_packet=64M' used above sets the maximum size of an SQL statement to be very large. In some circumstances, you may need to increase it further, especially if attachments are stored in the database.

Testing database encoding

See [Troubleshooting Character Encodings](#) for a number of tests you can run to ensure your database encoding is correct.

RELATED TOPICS:


[Character encodings in Confluence](#)
[Known Issues for MySQL](#)

Database Setup Guides






- [Database JDBC drivers](#)
- [Database Setup for Oracle](#)
- [Database Setup for SQL Server](#)
- [Database Setup For Any External Database](#)
- [Database Setup for PostgreSQL](#)
- [Database Setup For MySQL](#)

Database JDBC drivers

The JDBC drivers for all databases currently supported for Confluence are linked below. You will need to make the driver available to your application server, as described in the appropriate [setup guide](#).

 Please note, we bundle a number of JDBC drivers with Confluence, as shown below. You do not have to download or install the drivers for the relevant databases, **if you are using a direct JDBC connection**. If you are connecting via a datasource, you will still need to download and install the drivers manually.

JDBC Driver Download Links

Database	JDBC Drivers Bundled with Confluence?	JDBC Drivers
PostgreSQL		8.4-701.jdbc3 (note, the JDBC 3 driver will work under the 1.6 JVM. If you want to use the JDBC 4 driver, you can download it from the PostgreSQL website . However, we recommend that you use the bundled JDBC 3 driver.)
MySQL		5.1.11
Microsoft SQL Server		jTDS 1.2.2
Oracle		JDBC driver downloads (see Database Setup for Oracle for required JDBC driver versions)
DB2		JDBC drivers should be included with DB2, otherwise they can be downloaded from the IBM website

Database Setup for Oracle

This guide covers deploying the Confluence or Confluence WAR distributions with an Oracle database.



This database can only be set up by an Oracle database administrator (DBA)

If you are not a DBA, you should not attempt to set up this database.

Oracle has a history of being extremely difficult to set up. If you do not have access to an experienced Oracle DBA in your organisation, you

are recommended to select any free, scalable and easy-to-install alternative rather than proceeding with Oracle. Users evaluating Confluence are recommended to start with an alternative database and only consider migrating to Oracle after approval from their DBA. Atlassian's technical support for Oracle setup difficulties will also reflect the high minimum skill requirements for attempting an Oracle setup.

Database Setup Information

This setup guide must be used in conjunction with the list of [Known Issues For Oracle](#). Please review that page before continuing.

Schema Requirements

Confluence can be deployed to a schema in any Oracle instance.

Database Compatibility

Please refer to [Supported Platforms](#) for information about supported database versions. If your version of Oracle is not supported, please upgrade to a supported version before installing Confluence.

Check your database drivers, to see if you need an update.

- For Oracle 11.1, use the 10.2.0.4 or 11.1.0.7.0 driver (Java 6 ojdbc6.jar).
- For Oracle 11.2, use the 11.2.0.1.0 driver (Java 6 ojdbc6.jar).

Tip: search for the jar filename on the download site.

Check that your version of Oracle does not have any known issues:

Oracle Version	Oracle Driver	Issue	Solution
Any	Pre 10g	Driver incompatibilities	Upgrade to latest 10g drivers if compatible

You may be also interested in the [relevant JIRA documentation](#) to check the compatibility of your Oracle server and driver.

Deploying Confluence with Oracle

Complete the instructions for [installing Confluence](#), then **return to this document instead of proceeding** to the Confluence Setup Guide.

Database Preparation

Tailor these instructions to your particular database version:

1. Perform any necessary database or driver upgrades. Download the [latest compatible database drivers](#). See the [Oracle JDBC driver FAQ](#).
2. Create a Confluence user and grant the following permissions to the user:

```
grant connect to <user>;
grant resource to <user>;
```

Do **not** grant the database user the `select any table` permission, or it can cause problems with other schemas. See [CONF-3613](#) for a report.

Adding a Datasource to Tomcat

1. Open `<INSTALL>/conf/server.xml` for editing.
2. Locate the section `Host -> Context`

```
<Host name="localhost" debug="0" appBase="webapps" unpackWARs="true" autoDeploy="false">
  <Context path="" docBase="../confluence" debug="0" reloadable="true">
    <!-- Logger is deprecated in Tomcat 5.5. Logging configuration for Confluence
is specified in confluence/WEB-INF/classes/log4j.properties -->
    <Manager pathname="" />
  </Context>
</Host>
```

3. Paste in the Resource section provided, before Managers as shown:

```
<Host name="localhost" debug="0" appBase="webapps" unpackWARs="true" autoDeploy="false">
  <Context path="/" docBase="../confluence" debug="0" reloadable="true">
    <!-- Logger is deprecated in Tomcat 5.5. Logging configuration for Confluence
    is specified in confluence/WEB-INF/classes/log4j.properties -->

    <Resource
      name="jdbc/confluence"
      auth="Container"
      type="javax.sql.DataSource"
      driverClassName="oracle.jdbc.OracleDriver"
      url="jdbc:oracle:thin:@hostname:port:sid"
      username="<username>"
      password="<password>"
      connectionProperties="SetBigStringTryClob=true"
      maxActive="25"
      maxIdle="5"
      maxWait="10000"
    />

    <Manager pathname="" />
  </Context>
</Host>
```

4. Change the username and password to match the Oracle login.
5. Change url to match hostname, port and sid of the Oracle server. Note that sid stands for the Schema ID. For example:

```
jdbc:oracle:thin:@example.atlassian.com:1521:confluedb
```

For connecting to an Oracle RAC cluster, you'll need to edit the connection string using Oracle's connection syntax like this:

```
jdbc:oracle:thin:@(DESCRIPTION=
  (SDU=32768)
  (enable=broken)
  (LOAD_BALANCE=yes)
  (FAILOVER=yes)
  (ADDRESS=
    (PROTOCOL=TCP)
    (HOST=dbserver1.example.com)
    (PORT=1525))
  (ADDRESS=
    (PROTOCOL=TCP)
    (HOST=dbserver2.example.com)
    (PORT=1525))
  (CONNECT_DATA=
    (SERVICE_NAME=CONFDB)))
```

This example has been broken up over multiple lines for clarity, but it should be compacted into a single line.

6. If required, choose different maxActive and maxIdle values. These set how many total database connections will be allowed at one time, and how many will be kept open even when there is no database activity.

Configuring Confluence Datasource Access

Configure Confluence to use this datasource:

1. Edit the file <INSTALL>/confluence/WEB-INF/web.xml
2. Go to the end of the file and just before </web-app>, insert the following:

```
<resource-ref>
  <description>Connection Pool</description>
  <res-ref-name>jdbc/confluence</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

- Download the Oracle JDBC database drivers for your JDK version via the [Database JDBC drivers](#) page. We recommend using the thin drivers only. Copy the JAR file into `<confluence install>/WEB-INF/lib`. This directory path is potentially `<INSTALL>/lib` if Confluence is running off Apache Tomcat version 6 or above.

Running the Confluence Setup Wizard

Now Confluence is ready to attempt to connect to Oracle:

- Startup Confluence using `<INSTALL>/bin/startup.bat` or `<INSTALL>/bin/startup.sh`
- Insert your licence and select **External Database**.
- Select **Datasource Connection** using your Oracle version.
- Enter `java:comp/env/jdbc/confluence` for the name of the datasource.

Confluence should now deploy using the Oracle database specified. Please read this [comment](#) on Oracle database optimisation.

Oracle Configuration Tips

24-hour time format with Oracle 8i

We have received a report from a user that when an Oracle 8i database is configured to use 24-hour time as the default format, an exception like this may occur:

```
005-12-06 13:23:20 Loading root WebApplicationContext

2005-12-06 13:24:34 StandardContext[]: Exception sending context initialized event to listener
instance
of class com.atlassian.confluence.util.ConfluenceContextLoaderListener

org.springframework.beans.factory.BeanCreationException: Error creating bean with name
'userAccessor' defined in class path resource [applicationContext.xml]:
Can't resolve reference to bean 'userAccessorTarget' while setting property 'target';
nested exception is org.springframework.beans.factory.BeanCreationException: Error creating
bean with name 'userAccessorTarget' defined in class path
resource [applicationContext.xml]: Can't resolve reference to bean 'spacePermissionManager'
while setting property 'spacePermissionManager';
nested exception is org.springframework.beans.factory.BeanCreationException: Error creating
bean with name 'spacePermissionManager' defined in class path resource [securityContext.xml]:
Can't resolve reference to bean 'spacePermissionManagerTarget' while setting property 'target';
nested exception is org.springframework.beans.factory.BeanCreationException:
Error creating bean with name 'spacePermissionManagerTarget' defined in class path resource
[securityContext.xml]: Initialization of bean failed;
nested exception is org.springframework.jdbc.UncategorizedSQLException: (Hibernate operation):
encountered SQLException [Cannot create PoolableConnectionFactory];
nested exception is org.apache.commons.dbcp.SQLNestedException: Cannot create
PoolableConnectionFactory

...

org.apache.commons.dbcp.SQLNestedException: Cannot create PoolableConnectionFactory, cause:

java.sql.SQLException: ORA-00604: error occurred at recursive SQL level 1

ORA-12705: invalid or unknown NLS parameter value specified
```

One symptom of this problem is that Confluence may refuse to start after midday.

The workaround is to go to 'General Configuration' and set the default time format to "HH:mm".

RELATED TOPICS

[Known Issues For Oracle](#)

Database Setup for SQL Server

Use this guide in conjunction with the more general [Database Setup Guide for Any Database](#). These instructions add some reference notes specific to SQL Server.

- Review the [known issues for SQL Server](#).
- Identify which character encoding to use. To do this, check the encoding currently used by your [application server](#) and [Confluence](#).

- All three must use compatible encoding. For example, the default SQL Server encoding of USC-2 is compatible with UTF-8.
3. Create a new database (as an SQL administrator). If you set your application server and Confluence to use an encoding incompatible with USC-2, specify that character encoding for the database.
 4. Set the default collation for the database to be 'SQL_Latin1_General_CP1_CS_AS' (**case sensitive**). You can do this by issuing the following SQL query:

```
ALTER DATABASE <database_name> COLLATE SQL_Latin1_General_CP1_CS_AS
```

5. Configure the database to use the isolation level, Read Committed with Row Versioning. You can do this by issuing the following SQL query:

```
ALTER DATABASE <database_name>
SET READ_COMMITTED_SNAPSHOT ON
WITH ROLLBACK IMMEDIATE;
```

6. Create a new SQL user account for Confluence (as an SQL administrator). Provide full create, read and write permissions for the database tables. Please note, Confluence must be able to create its own schema.
7. Install the database drivers, if needed:



SQL Server JDBC Drivers bundled with Confluence

The JDBC drivers for this database are bundled with Confluence. You do not have to download or install any JDBC drivers to use this database with Confluence, **if you are using a direct JDBC connection**. If you are connecting via a datasource, you will still need to download and install the drivers manually. See [Database JDBC drivers](#) for more information on the bundled JDBC drivers.

- If you're not sure which connection you're using, it's most likely JDBC. A JNDI resource must be configured manually, as described in [Configuring a MySQL Datasource in Apache Tomcat](#).
- If you are configuring a datasource to connect to your MS SQL server database, you will need to place the jar file in `<confluence install>/WEB-INF/lib` (for Confluence 2.10 onwards) or `<confluence install>/common/lib` (for earlier versions). Information and links to the appropriate database drivers are available on [Database JDBC drivers](#). You may also find this page helpful: <http://jtds.sourceforge.net/faq.html>
- 8. Start Confluence and visit the home URL (e.g. <http://localhost:8090>) to start the Confluence Setup Wizard and select a Custom Install, insert the relevant connection information.
 - When prompted for a **driver class name** in the database setup step enter:

```
net.sourceforge.jtds.jdbc.Driver
```

- When prompted for the **jdbc url**, the format to use is:

```
jdbc:jtds:sqlserver://<server>:<port>/<database>
```

Configuring a SQL Server Datasource in Apache Tomcat

This page contains instructions on how to set up an SQL Server datasource connection for Confluence or Confluence EAR/WAR.

On this page:

- [1. Install the Driver](#)
- [2. Shut down Tomcat](#)
- [3. Configure Tomcat](#)
- [4. Configure the Confluence web application](#)
- [5. Configure Confluence](#)

1. Install the Driver

1. Download the latest SQLServer JTDS drivers from <http://sourceforge.net/projects/jtds/files/>.
2. After unpacking the file you have downloaded, you'll find a file called something like `jtds-1.2.5.jar` (whatever is the latest version).
3. Copy this file into the `common/lib` directory of your Tomcat installation. Be aware that this directory may be just `lib` for Tomcat version 6 and beyond (i.e. `<tomcat-install>/lib` rather than `<tomcat-install>/common/lib`).



Alternatively you can get the driver from `/confluence/WEB-INF/lib/jtds-1.2.2.jar` and move it into the `common/lib` directory of your Tomcat installation.

2. Shut down Tomcat

1. Run `bin/shutdown.sh` or `bin/shutdown.bat` to bring Tomcat down while you are making these changes.



Make a backup of your `<CONFLUENCE_HOME>/confluence.cfg.xml` file and your `<CONFLUENCE_INSTALL>/conf/server.xml` file so you can easily revert should there be a problem.

3. Configure Tomcat

1. Firstly, you need to edit `<confluence install>/conf/server.xml` and find the following lines:

```
<Context path="" docBase="../confluence" debug="0" reloadable="true">
  <!-- Logger is deprecated in Tomcat 5.5. Logging configuration for Confluence is
  specified in confluence/WEB-INF/classes/log4j.properties -->
```

2. Within the Context tags, directly after the opening `<Context.../>` line, insert the DataSource Resource tag:

```
<Resource name="jdbc/confluence" auth="Container" type="javax.sql.DataSource"
  username="yourDatabaseUser"
  password="yourDatabasePassword"
  driverClassName="net.sourceforge.jtds.jdbc.Driver"
  url="jdbc:jtds:sqlserver://localhost:1433/yourDatabaseName"
  maxActive="20"
  maxIdle="10"
  validationQuery="select 1" />
```

- Replace the username and password parameters with the correct values for your database
- In the url parameter, replace the word 'yourDatabaseName' with the name of the database your confluence data will be stored in.



Why is the validationQuery element needed?

When a database server reboots, or there is a network failure, all the connections in the connection pool are broken and this normally requires a Application Server reboot.

However, the Commons DBCP (Database Connection Pool) which is used by the Tomcat application server can validate connections before issuing them by running a simple SQL query, and if a broken connection is detected, a new one is created to replace it. To do this, you will need to set the "validationQuery" option on the database connection pool.



If switching from a direct JDBC connection to datasource, you can find the above details in your `<CONFLUENCE_HOME>/confluence.cfg.xml` file.



The configuration properties for Tomcat's standard data source resource factory (`org.apache.tomcat.dbcp.dbcp.BasicDataSourceFactory`) are as follows:

- `driverClassName` — Fully qualified Java class name of the JDBC driver to be used.
- `maxActive` — The maximum number of active instances that can be allocated from this pool at the same time.
- `maxIdle` — The maximum number of connections that can sit idle in this pool at the same time.
- `maxWait` — The maximum number of milliseconds that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception.
- `password` — Database password to be passed to our JDBC driver.
- `url` — Connection URL to be passed to our JDBC driver. (For backwards compatibility, the property `driverName` is also recognized.)
- `user` — Database username to be passed to our JDBC driver.
- `validationQuery` — SQL query that can be used by the pool to validate connections before they are returned to the application. If specified, this query MUST be an SQL SELECT statement that returns at least one row.

4. Configure the Confluence web application

1. Edit `/confluence/WEB-INF/web.xml` in your confluence installation

2. Go to the end of the file and just before `</web-app>`, insert the following:

```
<resource-ref>
  <description>Connection Pool</description>
  <res-ref-name>jdbc/confluence</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

5. Configure Confluence

- **If you have not yet set up Confluence**
 1. Follow the steps in the [Confluence Setup Guide](#)
 2. In the [Database Setup](#) section, choose the "Datasource Connection" option.
 3. Set the JNDI name to `java:comp/env/jdbc/confluence`
 4. Set the Database dialect to SQL Server.
- **If you are changing an existing Confluence installation over to using a Tomcat datasource**
 1. Edit the `<confluence home>/confluence.cfg.xml` file
 2. Delete any line that contains a property that begins with `hibernate`.
 3. Insert the following at the start of the `<properties>` section.

```
<property name="hibernate.setup">true</property>
<property
name="hibernate.dialect">net.sf.hibernate.dialect.SQLServerInt1Dialect</property>
<property
name="hibernate.connection.datasource">java:comp/env/jdbc/confluence</property>
```

4. Restart Confluence.

RELATED TOPICS

[Configuring a MySQL Datasource in Apache Tomcat](#)

Database Setup For Any External Database

If you are using Confluence in a production environment, data should be stored in an external database. The embedded database is bundled for evaluation purposes and does not offer full transactional integrity in the event of sudden power loss.

This document provides instructions for setting up Confluence for use with a production-ready database. It covers both migration from an evaluation installation of Confluence and installation of an empty database during initial setup. The following specific database guides have additional information:

- [PostgreSQL Guide](#)
- [MySQL Guide](#)
- [Oracle Guide](#)
- [SQL Server](#)

Preparation

Install the following on the Confluence server:

- Database administration tool, for example [DBVisualizer](#)
- JDBC database drivers
- The database server (unless accessed remotely)

The instructions refer to two particular directories:

- The *<Confluence Installation Directory>* is the directory where you unpacked the Confluence download.
- The *<Confluence Home Directory>* is the directory where Confluence stores its data, which you set by editing the `confluence-init.properties` file in *Confluence Installation Directory/confluence/WEB-INF/classes*.

Database Setup

Create the schema and setup permissions:

1. Visit the [Database Configuration](#) page to review any known issues and database setup for your database.
2. Create a new schema using the [correct database encoding](#).
3. Create a user with full read/write access to the Confluence schema, including the ability to create tables.
4. If the database only permits users to log in from approved hosts (e.g. `localhost`), grant database access permission for the Confluence server.

5. If the database is hosted remotely to the Confluence server, set up any firewall permissions.
6. Test the connection by using the database administration tool installed on the Confluence server to log in to the database.

Migration From an Evaluation Instance of Confluence

Continue here if you are migrating from an evaluation instance with the built-in database. If you are installing Confluence for the first time, continue [below](#).

Create Backups

To keep any existing Confluence content:

1. If you are already using an external database, use your database administration tool to create a full database backup.
2. Manually create an XML backup of Confluence under '**Administration**' -> '**Backup & Restore**'. If you have less than 100MB of attachments, check '**Backup attachments**' when creating the backup. If you have over 100MB of attachments, you should not check the 'Backup attachments' and instead you should manually copy the `/attachments` directory in your Confluence home to a backup location. This attachments directory will later be copied into the new home directory.
3. Download the backup file to a backup location.

Database Connection Setup

Set up Confluence's database connection:

1. Stop Confluence if it is already running.
2. The JDBC database drivers for your database must be available to the application server. You can skip this step if the drivers are already loaded.
 - a. Copy the database driver JAR file into the `lib` directory. In Confluence this directory is `/confluence/WEB-INF/lib`. Other application servers will use a different path.
 - b. If the application server does not support dynamic library loading, stop your application server.
3. Create a new Confluence home directory.
4. Open the `WEB-INF/classes/confluence-init.properties` file in your Confluence installation and change the `confluence.home` property to point to this new Confluence home directory.
5. Start up Confluence. Refer to the platform-specific installation instructions to learn how. You should be presented with the **Confluence setup wizard**. Enter your license information.
6. Select '**Custom install**'.
7. Select a database from the drop down list.
8. Select `Direct JDBC` and then enter the username, password and database driver of the new database.
9. If you created a Confluence backup earlier and wish to restore it, import it into Confluence now.
10. Once the wizard is complete, if you did not check the 'Backup attachments', copy the backed up `/attachments` directory into the new Confluence home.

RELATED TOPICS

[Troubleshooting External Database Connections](#)

Database Setup for PostgreSQL

This document provides instructions for setting up Confluence for use with a PostgreSQL database. Please check the [Known Issues for PostgreSQL](#) before you start.

On this page:

- [1. Install PostgreSQL](#)
- [2. Create a User and a Database](#)
- [3. Configure Confluence to use the PostgreSQL Database](#)
- [Troubleshooting](#)

1. Install PostgreSQL

To install PostgreSQL,

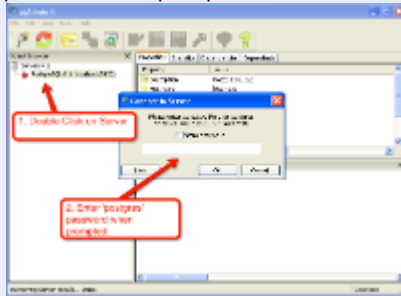
1. Download the database software and installer from the [PostgreSQL download site](#) and save it to your desktop. Choose the package that matches your operating system. Where available, choose the **One Click Installer**. These instructions assume you will use the One Click Installer. For example:
 - [PostgreSQL One Click Installer for Windows](#).
 - [PostgreSQL One Click Installer for Linux](#).
 - [PostgreSQL One Click Installer for Mac OS X](#).
2. Run the installer. Please note the following information when installing PostgreSQL:
 - The **password** that you are prompted to provide during the installation process is for the 'postgres' account, which is the db root level account.
 - The **default port** for PostgreSQL is 5432. If you decide to change the default port, please ensure that your new port number does not conflict with any services running on that port. You will also need to remember to update all further mentions of db port.
 - Choose the **locale** that best fits your geographic location, when prompted to enter a locale.
 - Do not launch **Stack Builder** at the completion of the installer.
3. PostgreSQL is now installed on your machine.

2. Create a User and a Database

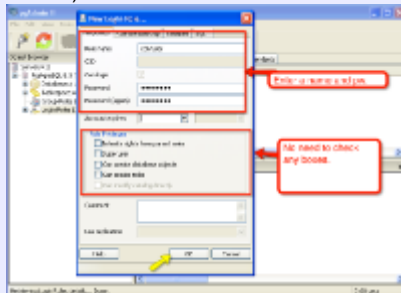
 All screenshots below are taken from a PostgreSQL configuration on a Windows machine.

To create a PostgreSQL user and database,

1. Start the 'pgAdmin III' administration tool on your machine. The pgAdmin III administration console will display. The database user and database that will be used by Confluence are created via the 'pgAdmin III' tool.
2. Connect to the PostgreSQL server (e.g. double-click on the server name in the object browser). Enter a 'postgres' password when prompted.

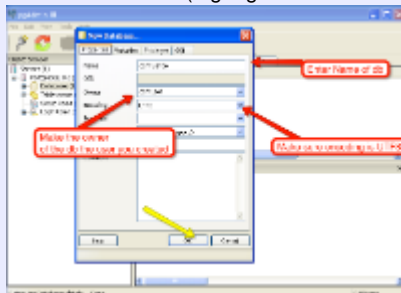


3. Create a new user, i.e. login role (e.g. right-click click 'Login Roles' in the object browser and select 'New Login Role...'):



- Enter a name and password for the new user.
- Do not select any role privileges.

4. Create a database (e.g. right-click 'Databases' and select 'New Database...'):



- Enter a name for the new database.
- Set the owner of the database to the user you created in the previous step.
- Select 'UTF8' for 'Encoding'.



Creating a User and Database via Linux command-line

If you are on Linux and do not have the above pgAdmin III administration tool, you can use the command line interface instead. Assuming that you are using the default installation directory of `/opt/PostgreSQL/8.3/bin/`, enter the following commands:

```
sudo -s -H -u postgres
# Create the Confluence user:
/opt/PostgreSQL/8.3/bin/createuser -S -d -r -P -E confuser
# Create the Confluence database:
/opt/PostgreSQL/8.3/bin/createdb -O confuser confluence
exit
```

3. Configure Confluence to use the PostgreSQL Database

Once you have installed and set up PostgreSQL, you will need to configure Confluence to use the PostgreSQL database.

To configure Confluence to use PostgreSQL,

1. Install Confluence, if you haven't done so already. Ensure that you download the [Confluence distribution](#), not the evaluation installer.
2. Ensure that Confluence is stopped (for example, by ensuring that the [application server](#) or service which is running Confluence has been stopped or terminated).
3. Install the database drivers, if needed:



PostgreSQL JDBC Drivers bundled with Confluence

The JDBC drivers for this database are bundled with Confluence. You do not have to download or install any JDBC drivers to use this database with Confluence, **if you are using a direct JDBC connection**. If you are connecting via a datasource, you will still need to download and install the drivers manually. See [Database JDBC drivers](#) for more information on the bundled JDBC drivers.

- If you're not sure which connection you're using, it's most likely JDBC. A JNDI resource must be configured manually, as described in [Configuring a MySQL Datasource in Apache Tomcat](#).

Note: Confluence only bundles the JDBC 3 driver which will work under the 1.6 JVM. However, if you are using Java 6 and want to use the JDBC 4 driver, you can download it via [Database JDBC drivers](#) and install it as described below. You will need to remove the existing PostgreSQL JDBC 3 driver (e.g. postgresql-8.4-701.jdbc3), if you do want to use the JDBC 4 driver.

- If you are configuring a datasource to connect to your PostgreSQL database, you will need to place the jar file in `<confluence install>/WEB-INF/lib` (for Confluence 2.10 onwards) or `<confluence install>/common/lib` (for earlier versions). Information and links to the appropriate database drivers are available on [Database JDBC drivers](#).



Windows renames .jar extensions to .zip! Just rename it back to .jar. You'll have to set your folder options to view hidden file extensions if you can't rename it without changing the file type (Tools >> Folder Options >> View >> Uncheck "Hide Extensions for known file types."

4. Start Confluence and after entering your license code on the '**Confluence Setup Wizard**' page, click '**Custom Installation**'. The '**Choose a Database Configuration**' page will display.
5. Select '**PostgreSQL**' and click '**External Database**'. The '**Configure Database**' page will display.
6. Choose your desired database connection method (please note that if you choose to connect via datasource, you will need to install the appropriate database drivers as described in the previous step).
7. Enter your PostgreSQL database setup details (as defined in the previous step [above](#)):



Connecting to an SSL Database

Simply add `ssl=true` parameter in the Database URL, for example:

```
jdbc:postgresql://localhost:5432/confluence?ssl=true
```



If the server that is hosting the PostgreSQL database is not the same server as Confluence, then please ensure that the confluence server can contact the database server and also refer to the PostgreSQL documentation on how to set up [pg_hba.conf](#). If the `pg_hba.conf` file is not set properly, remote communication to the PostgreSQL server will fail.



Running SQL Queries

For ongoing maintenance of your server, you can continue to use PGAdmin as your SQL browser.

Troubleshooting

- [Known Issues for PostgreSQL](#) contains common issues encountered when setting up your PostgreSQL database to work with Confluence.
- If you are unable to connect to the database from Confluence and they are on different machines, most likely you have a firewall in between the two machines or your `pg_hba.conf` file is misconfigured. Verify that your firewall is set to allow connections through 5432 or double check your [hba configuration](#)
- If Confluence is complaining that it's missing a class file, you might have forgotten to place the jdbc driver in the `WEB-INF/lib` folder or possibly have placed it in the wrong folder.
- If none of the above describes your issue, please create a support ticket at <http://support.atlassian.com> and be sure to include your logs (found in `confluence-install/logs` and `confluence-data/logs`).

Configuring a PostgreSQL Datasource in Apache Tomcat

This page contains instructions on how to set up an PostgreSQL datasource connection for Confluence or Confluence EAR/WAR.

On this page:

- [1. Install the Driver](#)
- [2. Shut down Tomcat](#)
- [3. Configure Tomcat](#)
- [4. Configure the Confluence web application](#)
- [5. Configure Confluence](#)

1. Install the Driver

1. Download the PostgreSQL driver from <http://jdbc.postgresql.org/download.html>.
2. Copy this file into the `common/lib` directory of your Tomcat installation. Be aware that this directory may be just `lib` for Tomcat version 6 and beyond (i.e. `<tomcat-install>/lib` rather than `<tomcat-install>/common/lib`).



If you are using Confluence 3.2.0 or later you can get the driver from `/confluence/WEB-INF/lib/postgresql-8.4-701.jdbc3.jar` and move it into the `common/lib` directory of your Tomcat installation.

2. Shut down Tomcat

1. Run `bin/shutdown.sh` or `bin/shutdown.bat` to bring Tomcat down while you are making these changes.



Make a backup of your `<CONFLUENCE_HOME>/confluence.cfg.xml` file and your `<CONFLUENCE_INSTALL>/conf/server.xml` file so you can easily revert should there be a problem.

3. Configure Tomcat

1. Firstly, you need to edit `<confluence install>/conf/server.xml` and find the following lines:

```
<Context path="" docBase="../confluence" debug="0" reloadable="true">
  <!-- Logger is deprecated in Tomcat 5.5. Logging configuration for Confluence is
  specified in confluence/WEB-INF/classes/log4j.properties -->
```

2. Within the Context tags, directly after the opening `<Context...>` line, insert the DataSource Resource tag:

```
<Resource name="jdbc/confluence" auth="Container" type="javax.sql.DataSource"
  username="postgres"
  password="postgres"
  driverClassName="org.postgresql.Driver"
  url="jdbc:postgresql://localhost:5432/test"
  maxActive="20"
  maxIdle="10"
  validationQuery="select 1" />
```

- Replace the username and password parameters with the correct values for your database
- In the url parameter, replace the word 'yourDatabaseName' with the name of the database your confluence data will be stored in.



Why is the validationQuery element needed?

When a database server reboots, or there is a network failure, all the connections in the connection pool are broken and this normally requires a Application Server reboot.

However, the Commons DBCP (Database Connection Pool) which is used by the Tomcat application server can validate connections before issuing them by running a simple SQL query, and if a broken connection is detected, a new one is created to replace it. To do this, you will need to set the "validationQuery" option on the database connection pool.



If switching from a direct JDBC connection to datasource, you can find the above details in your `<CONFLUENCE_HOME>/confluence.cfg.xml` file.



The configuration properties for Tomcat's standard data source resource factory (org.apache.tomcat.dbcp.dbcp.BasicDataSourceFactory) are as follows:

- driverClassName — Fully qualified Java class name of the JDBC driver to be used.
- maxActive — The maximum number of active instances that can be allocated from this pool at the same time.
- maxIdle — The maximum number of connections that can sit idle in this pool at the same time.
- maxWait — The maximum number of milliseconds that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception.
- password — Database password to be passed to our JDBC driver.
- url — Connection URL to be passed to our JDBC driver. (For backwards compatibility, the property driverName is also recognized.)
- user — Database username to be passed to our JDBC driver.
- validationQuery — SQL query that can be used by the pool to validate connections before they are returned to the application. If specified, this query MUST be an SQL SELECT statement that returns at least one row.

4. Configure the Confluence web application

1. Edit `/confluence/WEB-INF/web.xml` in your confluence installation
2. Go to the end of the file and just before `</web-app>`, insert the following:

```
<resource-ref>
  <description>Connection Pool</description>
  <res-ref-name>jdbc/confluence</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

5. Configure Confluence

- If you have not yet set up Confluence
 1. Follow the steps in the [Confluence Setup Guide](#)
 2. In the [Database Setup](#) section, choose the "Datasource Connection" option.
 3. Set the JNDI name to `java:comp/env/jdbc/confluence`
 4. Set the Database dialect to Postgres.
- If you are changing an existing Confluence installation over to using a Tomcat datasource
 1. Edit the `<confluence home>/confluence.cfg.xml` file

2. Delete any line that contains a property that begins with hibernate.
3. Insert the following at the start of the <properties> section.

```
<property name="hibernate.setup"><![CDATA[true]]></property>
<property
name="hibernate.dialect"><![CDATA[net.sf.hibernate.dialect.PostgreSQLDialect]]></prope
name="hibernate.connection.datasource"><![CDATA[ java:comp/env/jdbc/confluence]]></prop
```

4. Restart Confluence.

RELATED TOPICS

[Configuring a MySQL Datasource in Apache Tomcat](#)

Database Setup For MySQL

This page provides instructions for installing Confluence and the open-source [MySQL database](#) on Microsoft Windows, as well as how to set up and configure MySQL to work with Confluence. Additional instructions are also provided for migrating across any existing Confluence database content. Use this guide in conjunction with the more general [Database Setup Guide for Any Database](#). These instructions add some important reference notes specific to MySQL.



Please note the following points:

- Throughout the instructions below, the `Confluence Installation Directory` refers to the directory where you extracted the Confluence zipped installer.
- The **MySQL Database Setup** procedure below will make modifications to your default MySQL Server settings. These modifications result in:
 1. The default collation (localisation) option being changed to `utf8_bin` (that is, case-sensitive binary UTF8), such that by default, all new database tables will be created with this type of case-sensitive collation.
 2. The default MySQL database storage engine being changed to 'InnoDB'.

On this page:

- [1. Install Confluence](#)
- [2. Install MySQL Server](#)
- [3. Set up your MySQL Database and User](#)
- (Optional) [4. Back Up Confluence Data](#)
- [5. Set Up your Database Connection](#)
- [Troubleshooting](#)
- [Related Documents](#)

1. Install Confluence


Install [Confluence](#) if you have not done so already. Ensure that you download [Confluence](#), not the evaluation installer.

2. Install MySQL Server


To install MySQL Server,


1. If you do not have an operational MySQL database server instance available, install 'MySQL Community Server' database server (version 5.0). The installation package can be downloaded from the [MySQL download page](#) or from the [version 5.0 download page](#). Instructions for installing the MySQL 5.0 database server on Windows can found on the 'Installing MySQL on Windows' page of the MySQL web site.

2. Run the 'MySQL Server Instance Config Wizard':

 If you intend to connect Confluence to an existing, operational MySQL database server instance, we strongly recommend that you reconfigure this database server instance by running through the Config Wizard and initially choosing the **Reconfigure Instance** option.

- a. At the start of the Config Wizard (or after having chosen **Reconfigure Instance**), choose **Detailed Configuration**.
- b. Choose the type of MySQL Server that best suits your hardware requirements. This will affect the MySQL Server's usage of memory, disk and CPU resources. Refer to the [relevant MySQL documentation](#) for further information.
- c. Choose **'Transactional Database Only'** for the database usage step. This ensures that your MySQL database will use InnoDB as its default storage engine.

 It is **highly recommended** that you only use the InnoDB storage engine with Confluence. Avoid using the MyISAM storage engine as this can lead to data corruption.
- d. Set the InnoDB Tablespace settings to your requirements. (The default settings are acceptable).
- e. Set the approximate number of concurrent connections permitted that best suits your Confluence usage requirements. You can use one of the presets if desired or enter a number manually. Refer to the [relevant MySQL documentation](#) for further information.
- f. For the networking options, ensure the **'Enable TCP/IP Networking'** and **'Enable Strict Mode'** options are selected (default). Refer to the MySQL documentation on setting the [networking](#) and [server SQL modes](#) for further information.
- g. For the MySQL server instance's default character set option, choose **'Best Support For Multilingualism'** (i.e. UTF-8).
- h. For the Windows configuration option, choose whether or not to install the MySQL Server as a Windows Service. If your hardware is going to be used as a dedicated MySQL Server, you may wish to choose the **'Install As Windows Service'** (and **Launch the MySQL Server automatically**) options. Refer to the [relevant MySQL documentation](#) for further information.

 If you choose not to install the MySQL Server as a Windows Service, you will need to ensure that the database service has been started before running Confluence.
- i. Finally, select the **'Modify Security Settings'** option to enter and set your MySQL Server (root) access password.

3. Install the database drivers, if needed:

**MySQL JDBC Drivers bundled with Confluence**

The JDBC drivers for this database are bundled with Confluence. You do not have to download or install any JDBC drivers to use this database with Confluence, **if you are using a direct JDBC connection**. If you are connecting via a datasource, you will still need to download and install the drivers manually. See [Database JDBC drivers](#) for more information on the bundled JDBC drivers.

- If you're not sure which connection you're using, it's most likely JDBC. A JNDI resource must be configured manually, as described in [Configuring a MySQL Datasource in Apache Tomcat](#).
- If you are configuring a datasource to connect to your MySQL database, you will need to place the jar file (i.e. `mysql-connector-java-5.x.y-bin.jar` where `x.y` depends on the driver version) in `<confluence install>/WEB-INF/lib` (for Confluence 2.10 onwards) or `<confluence install>/common/lib` (for earlier versions). Information and links to the appropriate database drivers are available on [Database JDBC drivers](#).

- Configure your MySQL server's settings by editing MySQL's `my.cnf` file (often named `my.ini` on Windows operating systems). Locate the `[mysqld]` section in the file add/modify the following parameters:

- Specify the default character set to be UTF-8:

```
[mysqld]
...
default-collation=utf8_bin
character-set-server=utf8
collation-server=utf8_bin
default-character-set=utf8
...
```

- Set the default storage engine to InnoDB:

```
[mysqld]
...
default-storage-engine=INNODB
...
```

- Specify/increase the value of `max_allowed_packet` to be at least 32M:

```
[mysqld]
...
max_allowed_packet=32M
...
```

(Refer to [MySQL Option Files](#) for detailed instructions on editing `my.cnf` and `my.ini`.)

- Restart your MySQL server for the changes to take effect:
 - On Windows, use the Service Manager to restart the service
 - On Linux, depending on your setup you will need to run either `/etc/init.d/mysqld stop`, `/etc/init.d/mysql stop` or `service mysqld stop` followed by the same command with "stop" replaced with "start"
 - On Mac OS X, run `sudo /Library/StartupItems/MySQLCOM/MySQLCOM restart`

3. Set up your MySQL Database and User

To create the database and user privileges,


- Run the `mysql` command as a MySQL super user. The default is "root" with a blank password.
- Create an empty Confluence database schema by running

```
CREATE DATABASE confluence;
```

- Create the Confluence database user by running

```
GRANT ALL PRIVILEGES ON confluence.* TO 'confluenceuser'@'localhost'
IDENTIFIED BY 'confluencepass';
```

Replace "confluenceuser" and "confluencepass" with a username and password of your choice. If Confluence is not running on the same server as your MySQL database server, replace "localhost" with the hostname or IP address of the Confluence server.

 To support international languages in Confluence, you should [verify](#) that the newly created database is using UTF-8 encoding and re-examine the JDBC URL settings (configured in Stage 3).

**For an existing database**

If you're using an existing database, confirm the Character Encoding by executing the query:
`SHOW VARIABLES LIKE 'character%';` and `SHOW VARIABLES LIKE 'collation%';`.
 The results should be UTF-8.

(Optional) 4. Back Up Confluence Data

This stage is only required if you have existing Confluence content you wish to transfer.

To back up your Confluence data,

1. Manually create an XML backup of Confluence under Administration -> Backup & Restore. If you have less than 100MB of attachments, check 'Backup attachments' when creating the backup. If you have over 100MB of attachments, you should not check the 'Backup attachments' and instead you should manually copy the /attachments directory in your Confluence home to another location. This attachments directory can then be copied into the new home directory as [described later](#).
2. Download the backup file to a backups folder.

5. Set Up your Database Connection

To set up your Confluence MySQL database connection or to switch to using this external database,

1. Ensure that Confluence is stopped (for example, by ensuring that the [application server](#) or service which is running Confluence has been stopped or terminated).
2. If you haven't started Confluence yet, you can skip this step. If you have set up Confluence with the built-in (HSQLDB) database, edit Confluence Installation Directory_/confluence/WEB-INF/classes/confluence-init.properties and change the confluence.homeproperty to point to a new directory. e.g. if you had

```
confluence.home=c:/confluencedata
```

You could change it to:

```
confluence.home=c:/confluencedata_mysql
```

This is your new Confluence Home Directory. (The name doesn't have to end in _mysql – that's just an example)

3. Start Confluence and set up the new configuration.
 - a. Enter your license key and click the **'Custom Installation'** button.
 - b. Under the 'External Database' heading, select **'MySQL'** from the dropdown list and click **'External Database'**.
 - c. On the next page, click **'Direct JDBC'**.
 - d. Enter `confluenceuser` in the User Name field, and the password you chose earlier in the Password field
 - e. Click the Next button. If you get the error message `Could not successfully test your database: Server connection failure during transaction. Due to underlying exception: 'java.sql.SQLException: Access denied for user 'confluenceuser'@'localhost' (using password: YES)'` verify that you have properly given the `confluenceuser` user all the right permissions when connecting from localhost.
 - f. (optional) If you previously [backed up your Confluence data](#), you can choose to restore it at the 'Load Content' page. Choose 'Restore From Backup', browse for the backup you created and restore it. Otherwise choose either the example or empty site as you wish.

Troubleshooting

- [Known Issues for MySQL](#) contains common issues encountered when setting up your MySQL database to work with Confluence.
- If Confluence is complaining that it's missing a class file, you might have forgotten to place the jdbc driver in the `WEB-INF/lib` folder or possibly have placed it in the wrong folder.
- If none of the above describes your issue, please create a support ticket at <http://support.atlassian.com> and be sure to include your

logs (found in `confluence-install/logs` and `confluence-data/logs`).

Related Documents

- [Configuring Database Character Encoding](#)
- [Known Issues for MySQL](#)

Configuring a MySQL Datasource in Apache Tomcat

This page tells you how to set up a MySQL datasource connection for Confluence or Confluence EAR/WAR.

On this page:

- [1. Shut down Tomcat](#)
- [2. Install the Drivers](#)
- [3. Configure Tomcat](#)
- [4. Configure the Confluence Web Application](#)
- [5. Configure Confluence](#)
- [6. Restart Confluence](#)

1. Shut down Tomcat

1. Run `bin/shutdown.sh` or `bin/shutdown.bat` to bring Tomcat down while you are making these changes.
2. Make a backup of your `<CONFLUENCE_HOME>/confluence.cfg.xml` file and your `<CONFLUENCE_INSTALL>/conf/server.xml` file so that you can easily revert if you have a problem.

2. Install the Drivers

1. Download the MySQL JDBC drivers from <http://www.mysql.com/downloads/api-jdbc-stable.html>.
2. After unpacking the file you have downloaded, you will find a file called something like `mysql-connector-java-3.0.10-stable-bin.jar`.
3. Copy this file into the `common/lib` directory of your Tomcat installation. Be aware that this directory may be just `lib` for Tomcat version 6 and beyond (i.e. `<tomcat-install>/lib` rather than `<tomcat-install>/common/lib`).

3. Configure Tomcat

1. If you are using the Confluence distribution, edit the `conf/server.xml` file in your Tomcat installation. If you are running your own Tomcat instance, edit the XML file where you declared the Confluence Context descriptor.
2. If editing `conf/server.xml`, find the following lines:

```
<Context path="" docBase="../confluence" debug="0" reloadable="true">
    <!-- Logger is deprecated in Tomcat 5.5. Logging configuration for
    Confluence is specified in confluence/WEB-INF/classes/log4j.properties -->
```

3. Within the Context tags, directly after the opening `<Context.../>` line, insert the DataSource Resource tag:

```
<Resource name="jdbc/confluence" auth="Container" type="javax.sql.DataSource"
    username="yourusername"
    password="yourpassword"
    driverClassName="com.mysql.jdbc.Driver"

    url="jdbc:mysql://localhost:3306/confluence?useUnicode=true&characterEncoding=utf8"
    maxActive="15"
    maxIdle="7"
    defaultTransactionIsolation="READ_COMMITTED"
    validationQuery="Select 1" />
```

- Replace the username and password parameters with the correct values for your database.
- In the url parameter, replace the word 'confluence' with the name of the database your Confluence data will be stored in.
- If you plan to use non-Latin characters, you will also need to add `"&useUnicode=true&characterEncoding=utf8"` on the end of the above URL. These options are not required for any database other than MySQL.

Notes

- If switching from a direct JDBC connection to datasource, you can find the above details in your

<CONFLUENCE_HOME>/confluence.cfg.xml file.

- Why is the validationQuery element needed? When a database server reboots, or there is a network failure, all the connections in the connection pool are broken and this normally requires an application server reboot.

However, the Commons DBCP (Database Connection Pool) which is used by the Tomcat application server can validate connections before issuing them by running a simple SQL query, and if a broken connection is detected, a new one is created to replace it. To do this, you will need to set the "validationQuery" option on the database connection pool.

- The configuration properties for Tomcat's standard data source resource factory (org.apache.tomcat.dbcp.dbcp.BasicDataSourceFactory) are as follows:
 - driverClassName – Fully qualified Java class name of the JDBC driver to be used.
 - maxActive – The maximum number of active instances that can be allocated from this pool at the same time.
 - maxIdle – The maximum number of connections that can sit idle in this pool at the same time.
 - maxWait – The maximum number of milliseconds that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception.
 - password – Database password to be passed to our JDBC driver.
 - url – Connection URL to be passed to our JDBC driver. (For backwards compatibility, the property driverName is also recognized.)
 - user – Database username to be passed to our JDBC driver.
 - validationQuery – SQL query that can be used by the pool to validate connections before they are returned to the application. If specified, this query MUST be an SQL SELECT statement that returns at least one row.

4. Configure the Confluence Web Application

1. Edit confluence/WEB-INF/web.xml in your Confluence installation.
2. Go to the end of the file and insert the following element just before </web-app>:

```
<resource-ref>
  <description>Connection Pool</description>
  <res-ref-name>jdbc/confluence</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

5. Configure Confluence

If you have not yet set up Confluence:

1. Follow the steps in the [Confluence Setup Guide](#)
2. In the [Database Setup](#) section, choose the "Datasource Connection" option.
3. Set the JNDI name to java:comp/env/jdbc/confluence
4. Set the Database dialect to MySQL.

If you are changing an existing Confluence installation over to using a Tomcat datasource:

1. Find your ConfluenceHome directory (see: [Confluence Home Directory](#) if you don't know where it is).
2. Edit the confluence.cfg.xml file
3. Delete any line that contains a property that begins with hibernate.
4. Insert the following at the start of the <properties>section.

```
<property name="hibernate.setup"><![CDATA[true]]></property>
  <property
name="hibernate.dialect"><![CDATA[net.sf.hibernate.dialect.MySQLDialect]]></property>
  <property
name="hibernate.connection.datasource"><![CDATA[java:comp/env/jdbc/confluence]]></property>
```

6. Restart Confluence

Run bin/startup.sh or bin/startup.bat to start Tomcat with the new settings.

RELATED TOPICS

[Database Setup For MySQL](#)

Creating Database Schema Manually

Database Schema Creation

This document provides information on how to find the SQL for database table creation.

Often, DBAs will require that table creation be done manually. To find the required SQL statements, you can use the script generated from our evaluation version's HSQLDB database.

- Install Confluence with no external database.
- Shut down Confluence.
- Find the file located in `<confluence-home>/database/confluencedb.script`.

The `confluence-home` directory is not your installation directory, its the directory specified in `confluence-init.properties`. For more information, see [Confluence Home Directory](#) and [Confluence Installation Directory](#).

To see some example SQL statements, click 'Tools' and select 'Attachments' on this wiki page.

To get Confluence to run against this established schema, configure the database with the normal procedure, then modify the `hibernate.connection.url`, `username` and `password` from `confluence-home/confluence.cfg.xml` or `server.xml`, depending on whether it's a direct jdbc or resource connection. This is described in [Migrating Confluence Between Servers](#).

RELATED TOPICS

[Database Configuration](#)

Migrate to Another Database

Limitations of Database Migration



The XML backup built into Confluence is not well suited for database migration for large data sets (see [Production Backup Strategy](#) for reference). If the procedures below do not work, use a commercial database migration tool.

Database Migration

This document outlines how to migrate your data from your existing database to another database. It is designed for migrating from an evaluation to a production database. Large data sets will require third party database migration tools. You should use this method when moving from the embedded DB to an external DB, or from one type of external DB to another (e.g. Oracle to Postgres).



If you are simply moving your DB from one server to another you can just change the JDBC URL in `<confluence.home>/confluence.cfg.xml` (if you are using a direct JDBC connection) or in the definition of your datasource (if you are connecting via a datasource).

There are two ways you can perform the migration:

1. Method one is the standard procedure.
2. For large installations of Confluence using version 2.2 or later: If the total size of attachments on your installation exceeds 500MB, use method two.

On this page:

- [Method One - Standard Procedure](#)
 - [Step One: Backing up your data](#)
 - [Step Two: Configuring the Confluence Home Directory](#)
 - [Step Three: Setting up the new database](#)
 - [Step Four: Setting up Confluence with the new database](#)
- [Method Two - For large installations](#)
 - [Step One: Backing up your data](#)
 - [Step Two: Configuring the Confluence Home Directory](#)
 - [Step Three: Moving your attachments](#)
 - [Step Four: Setting up new database](#)
 - [Step Five: Setting up Confluence with the new database](#)
- [A Note about Case Sensitivity in your Database](#)
- [Troubleshooting](#)

Method One - Standard Procedure

Step One: Backing up your data

1. Create a backup of your existing data. This is done from the **Administration Console**. Instructions on how to create a backup can be found [here](#).
2. Shut down and backup the [Confluence Home Directory](#).
3. If you are already using an external database, please make a backup of it using the utilities that were installed with it.



Note which [plugins](#) are currently installed/enabled, so that you can reinstate them later.

Step Two: Configuring the Confluence Home Directory


1. Create a new Confluence Home Directory. You can place this directory anywhere you like and give it a name of your choice.
2. Open WEB-INF/classes/confluence-init.properties file in your Confluence installation and change the confluence.home property to point to this new Confluence Home Directory.

Step Three: Setting up the new database

Perform the [database setup instructions](#) for your database.

Step Four: Setting up Confluence with the new database

If your databases are not already configured for Confluence, refer to [Database Configuration](#) to setup your database access.

1. Make sure that the JDBC drivers for your database are available to the application server. If you don't already have the JDBC driver, please download one from [here](#).
2. Make sure that your database is using a **case-sensitive** collation. Please refer to the section on case sensitivity [below](#) and see this issue for more details: [CONF-7917](#).
3. If you are running the Confluence distribution, copy your JDBC database driver (a .jar file), into the <confluence-install>/lib folder.
4. Start up Confluence. You will see the [Confluence Setup Wizard](#).
5. Select 'Custom Install'.
6. Select your database from the drop down list.
7. Select 'Direct JDBC' and then enter the details of the new database you want to migrate to.
8.  Read the documentation on the [Setup Wizard](#) for more detailed explanation.
8. When prompted, restore the contents of the backup you made in Step One into the new Confluence site

Your old Confluence data will now be imported to your new database.

Method Two - For large installations

Step One: Backing up your data

1. Before proceeding with these instructions please check that:
 - you are upgrading from at least Confluence version 2.2 and
 - your attachments are stored in the file system, and *not in your database*. (To migrate between attachment storage systems, please see [Attachment Storage Configuration](#))
2. From Confluence, go to Administration -> Backup & Restore and create a manual backup that **excludes** attachments.
3. Shut down and back up the [Confluence Home Directory](#).
4. If you are already using an external database, please make a backup of it using the utilities that were installed with it.

 These instructions will not work if either of the above is not true.

Step Two: Configuring the Confluence Home Directory

1. Create a new Confluence Home Directory. You can place this directory anywhere you like and give it a name of your choice.
2. Open WEB-INF/classes/confluence-init.properties file in your Confluence installation and change the confluence.home property to point to this new Confluence Home Directory.

Step Three: Moving your attachments

Move the contents of your attachments directory from your old Confluence Home to your new Confluence Home.


Step Four: Setting up new database

Perform the [database setup instructions](#) for your database.

Step Five: Setting up Confluence with the new database

If your databases are not already configured for Confluence, refer to [Database Configuration](#) to setup your database access.

1. Make sure that the JDBC drivers for your database are available to the application server. If you don't already have the JDBC driver, please download one from [here](#).
2. Make sure that your database is using a **case-sensitive** collation. Please refer to the section on case sensitivity [below](#) and see this issue for more details: [CONF-7917](#).
3. If you are running the Confluence distribution, copy your JDBC database driver (a .jar file), into the <confluence-install>/lib folder.
4. Start up Confluence. You will see the [Confluence Setup Wizard](#).
5. Select 'Custom Install'.

6. Select your database from the drop down list.
7. Select '**Direct JDBC**' and then enter the details of the new database you want to migrate to.
-  Read the documentation on the [Confluence Setup Wizard](#) for more detailed explanation.
8. When prompted, restore the contents of the backup you made in Step One into the new Confluence site.

A Note about Case Sensitivity in your Database

'Collation' refers to a set of rules that determine how data is sorted and compared. Case sensitivity is one aspect of collation. Other aspects include sensitivity to kana (Japanese script) and to width (single versus double byte characters).

Case sensitive or case insensitive collation – how should you create your Confluence database? What about when you are migrating your existing Confluence instance from one database to another?

Setting up a New Confluence Instance

For new Confluence instances, we recommend using case sensitive collation for your Confluence database. This is the default collation type used by many database systems.

Note: Even if the database is configured for case sensitive collation, Confluence reduces all usernames to lower case characters before storing them in the database. For example, this means that 'joebloggs', 'joeBloggs' and 'JoeBloggs' will be treated as the same username.

Migrating an Existing Confluence Instance to a Different Database

The default Confluence configuration uses case sensitive database collation. This is typical of databases created under default conditions. If you are migrating from this type of configuration to a new database, we recommend that the new database uses case sensitive collation. If you use case insensitive collation, you may encounter data integrity problems after migration (for example, via an XML import) if data stored within your original Confluence site required case sensitive distinctions.

Troubleshooting

If you're unable to restore your XML backup, consult our [Troubleshooting Guide](#).

RELATED TOPICS

Page: [Migrate to Another Database](#)

Page: [Confluence Configuration Guide](#)

 [Administrators Guide Home](#)  [Confluence Documentation Home](#)

The Embedded HSQLDB Database

The Confluence installation includes an embedded [HSQLDB](#) database, supplied for the purpose of **evaluating Confluence**.

If you are using the embedded database, the database files are stored in the `\database` folder under your [Confluence Home](#) directory. See also [Important Directories and Files](#).



Embedded Database is Not Suitable for Production Instances of Confluence

Production instances of Confluence should use an [external database](#). When using the default HSQLDB database, you run the risk of unrecoverable data loss due to not being [transaction safe](#).

- Corruption is occasionally encountered after sudden power loss and can usually be corrected using [this data recovery procedure](#).
- HSQLDB is still suitable for evaluation purposes, but the risk can only be eliminated by switching databases. External databases may also provide superior speed and scalability.

RELATED TOPICS

[Important Directories and Files](#)
[Database Configuration](#)

Troubleshooting the Embedded Database (HSQL DB)

java.sql.SQLException: User not found: SA

Also see <http://hsqldb.sourceforge.net/doc/guide/ch01.html#N101C2>.

HSQldb periodically must update its files to represent changes made in the database. In doing so, it must delete the current confluencedb.data file on the filesystem (beneath conf.home/database) and replace it with a new one.

If an administrator issues a shutdown on Confluence in this period, data can be lost, and is typically noticed by the error message, when starting Confluence up again, of 'User not found: SA'.

Users encountering this problem should seek to restore backups, contained in the backup directory beneath confluence.home. If daily backups have been turned off, and no other copy of data remains, the data is lost.



HSQldb should not be used as a production database. It is included for evaluation purposes only.

Hibernate logging

It can be useful to [enable detailed Hibernate logging](#) when debugging problems with HSQL.

Connecting to the Embedded Database

Connecting to the Embedded HSQL Database can be quite difficult. You may need to connect to the database to retrieve information, or for troubleshooting purposes.

Please follow the instructions on how to you can [connect to the embedded HSQL Database](#) using the free Database Administration Tool DBVisualizer.

Connecting to HSQldb using DBVisualizer

The purpose of this guide is to walk you through connecting to Confluence's embedded [Hypersonic SQL Database](#) using the Database Administration tool DBVisualizer.

Below are step by step instructions on how to Configure DBVisualizer and connect it to HSQldb.

Prerequisites

1. Download and install the latest copy of [DBVisualizer](#).
2. You will also need to download a copy (preferably the latest version) of [HSQldb](#)
3. Extract the contents of the HSQldb archive
4. Ensure that Confluence is not running.

Connection Procedure

Please ensure that you read and follow the instructions below carefully.



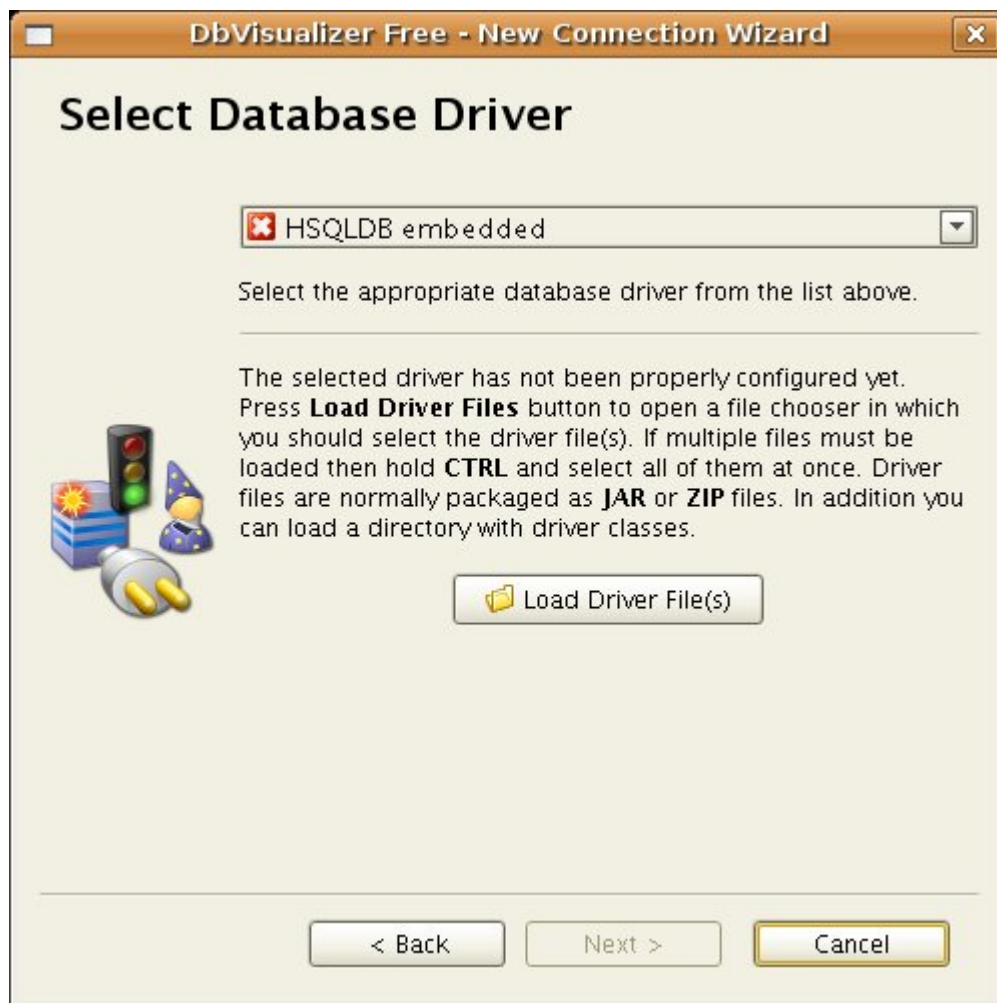
Remember to backup your <confluence-home>/database folder before attempting any modifications

1. Enter Connection Name



1. Click on the icon highlighted in **Red**
2. Enter an identifiable name for the connection. e.g. conf2.5.4-std

2. Select JDBC Driver

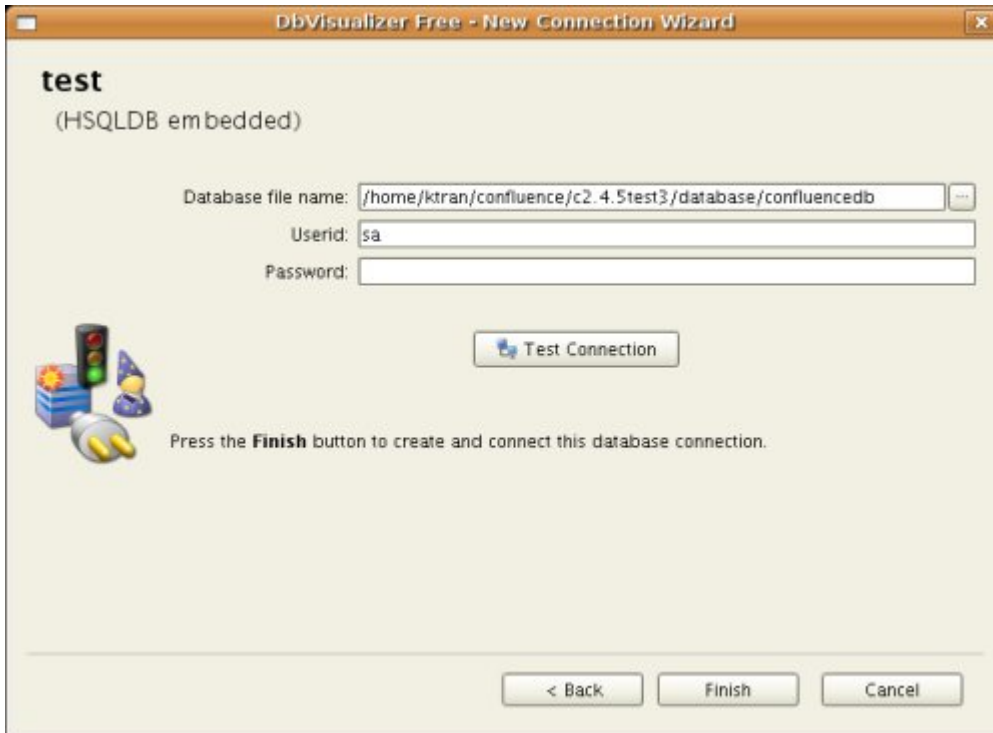


1. From the drop down list select **HSQLDB Embedded**
2. Click on Load Driver Files
3. Browse to directory where the HSQLDB.jar file is located. Confluence bundles this and it can be found at
 <confluence-installation>/confluence/WEB-INF/lib/hsqldb-*.jar.

3. Select Database Path

1. Browse to your <Confluence-Home> directory
2. Open the **Database** folder
3. Select the confluencedb.properties file

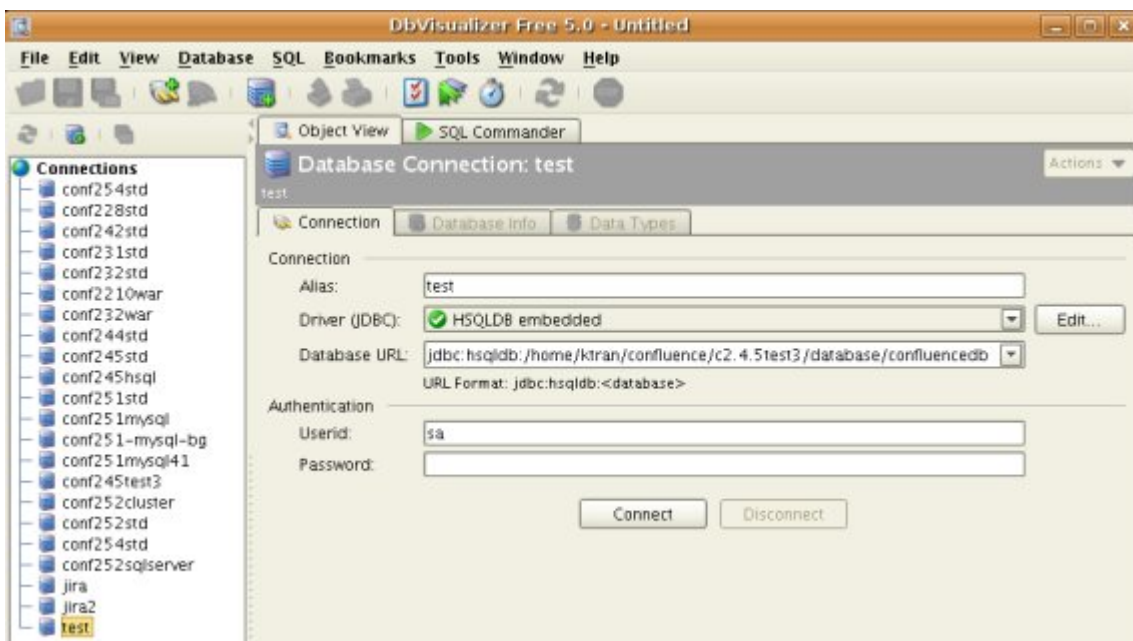
4. Enter Connection Details



1. Remove the ".properties" from the end of confluencedb
2. Type in sa for the username
3. Leave the password field blank

refer to the example screenshot above if you are unsure

5. Connect to embedded Database



1. Click on Test Connection to verify that the details are correct.
2. Click on "Finish" to complete the setup
3. Select the connection from the list on the left hand side.
4. You can now click on "Connect" to connect to the embedded database.

HSQL database manager

Alternatively, you can use [HSQLDB's database manager](#). Just copy the value of `hibernate.connection.url` in `confluence.cfg.xml` as the URL and you're good to go.

Related Topics

Below is a diagram of the Table References in Confluence (2.5.4).

This may be useful for Database Administrators that need to manually create the Database tables.



Troubleshooting External Database Connections

A common administration issue when configuring Confluence is identifying database connectivity problems. This page tells you about a helper utility, in the form of a JSP page, that can help you to isolate database connectivity issues. It checks whether you can connect to a database with your application server. If your application server cannot connect to the database, Confluence will not be able to connect to the database either.

You can use this utility to:

- Check that your application server can successfully query your database, either via immediate JDBC connectivity or a datasource in the context of your application server.
- Pinpoint problems in your configuration which may occur if the above is failing.

This is what the utility does:

- Check that a JDBC driver can be loaded into memory and view what is already loaded.
- Connect to a JDBC URL and do a 'select 1' from the database.
- Find a DataSource in the JNDI environment and do the above.
- View the System classpath (to ensure that the JDBC JAR file is there).

If you have already set up Confluence completely


1. Download the attached `testdatabase.jsp` to your `<confluence-install>\confluence` directory.
2. Restart Confluence
3. Go to <http://MY-CONF-SERVER:MY-CONF-PORT/testdatabase.jsp>.
4. Check that your database driver is loaded into memory. If not, check the system classpath for the JDBC driver file, and that the driver is in the `<confluence-install>\lib` directory (for Confluence version 2.10 onwards) or `<confluence-install>\common\lib` (for earlier versions). Here are some [instructions](#).
5. Enter the DB settings Confluence is using and test the database. If an error appears, check that the db service is running, the location matches, and that any users specified actually exist with the right login and permissions. You may be able to find a workaround by Googling the error.

1. Record the DB settings you are using for your direct JDBC or datasource connection in the 'Configure Database' step of your setup.
2. Download the attached [testdatabase.jsp](#) to your <confluence-install>\confluence directory.
3. Rename your <confluence-install>\confluence\WEB-INF\web.xml file to backup_web.xml. This disables redirection.

- Restart Confluence.
- Go to <http://MY-CONF-SERVER:MY-CONF-PORT/testdatabase.jsp>.
- Check that your database driver is loaded into memory. If not, check the system classpath for the JDBC driver file, and that the driver is in the `<confluence-install>\common\lib` directory as described in these [instructions](#).
- Enter the DB settings you recorded and test the database. If an error appears, check that the db service is running, the location matches, and that any users specified actually exist with the right login and permissions. You may be able to find a workaround by Googling the error.
- After correcting the error, rename `<confluence-install>\confluence\WEB-INF\backup web.xml` back to `web.xml`.

Notes

If you use this utility, please let us know ways in which we could improve it or leave helpful hints for others here.

 For a comprehensive set of database instructions that might be helpful for troubleshooting, please refer to the following links:

- [PostgreSQL](#)
- [MySQL](#)

Requesting Technical Support

If you are still stuck after attempting the suggestions above, [lodge a free technical support request](#) with information on your database setup.

Configuring database query timeout

If database queries are taking too long to perform, and your application is becoming unresponsive, you can configure a timeout for database queries. There is no default timeout in Confluence.

To configure a database query timeout, do the following on your test server:


- Shut down Confluence.
- Extract `databaseSubsystemContext.xml` from the `confluence-x.x.x.jar` that is in `confluence/WEB-INF/lib/`, and put a copy in `confluence/WEB-INF/classes/`.
- Edit `confluence/WEB-INF/classes/databaseSubsystemContext.xml` to add the `defaultTimeout` property to the "transactionManager" bean:

```
<bean id="transactionManager"
class="org.springframework.orm.hibernate.HibernateTransactionManager">
  <property name="sessionFactory">
    <ref bean="sessionFactory"/>
  </property>
  <property name="defaultTimeout" value="120"/>
</bean>
```

The timeout is measured in seconds and will forcibly abort queries that take longer than this. In some cases, these errors are not handled gracefully by Confluence and will result in the user seeing the Confluence error page.

- Start Confluence.

Once the timeout is working properly in your test environment, migrate the configuration change to Confluence.

 You will need to reapply these changes when upgrading Confluence, as the original `databaseSubsystemContext.xml` file changes from version to version.

Improving Database Performance

Diagnosis

Use native database tools to assess the impact of your database. If you'd like to check what Confluence is doing from its side, you can [enable sql login](#). If you analyze [thread dumps](#), as this is done in general [Troubleshooting Confluence Hanging or Crashing](#) guide, you may find the kinds of threads like this:

```
"http-8080-Processor150" daemon prio=1 tid=0x08543368 nid=0x11aa in Object.wait()
[0x665a4000..0x665a51b0]
at java.lang.Object.wait(Native Method)
- waiting on <0x83140488> (a com.mchange.v2.resourcepool.BasicResourcePool)
at com.mchange.v2.resourcepool.BasicResourcePool.awaitAcquire(BasicResourcePool.java:968)
at com.mchange.v2.resourcepool.BasicResourcePool.checkoutResource(BasicResourcePool.java:208)
- locked <0x83140488> (a com.mchange.v2.resourcepool.BasicResourcePool)
```

These threads are waiting for a database connection. It could be that the database is not performing optimally, or it may just need tuning for allowing more connection threads. Both are discussed below.

Upgrade your Database and Drivers

SQL Server 2000, Oracle 9i, and MySQL with 3.1 drivers are among some of the issues with database performance. Ensure you are using updated versions of databases and their drivers.

Upgrade your hardware

Atlassian does not offer specific recommendations on hardware for database performance. Use good judgment and native OS and database tools for your assessment.

Ensure you have the Latest Database Indices

Confluence has improved database performance over time. You'll want to make sure you have all the latest, if you're getting hung threads waiting for db connections.

Confluence 2.10 or Manual .ddl Indices

With 2.10 and later, Confluence includes database indices bundled. Confluence 2.10 [automatically creates the necessary database indexes](#) when you upgrade. If you are not on 2.10, you may have run the ddl manually during the upgrade process. To check, you can look against [these](#).

Additional Indices not Included in 2.10

- One import db index is the **lower case page title index**. Prior to Confluence 3.0, querying for a page by title and space key can take a long time due to table scans necessary on a lowercase *where* clause. On most databases it is possible to add a lowercase index on these columns that helps with performance. See [Creating a Lowercase Page Title Index](#) for instructions on how to do this. Prior to 2.10, apply lowercase title indexes (all Confluence versions).
- The compound database index for the ATTACHMENTDATA table is described in [CONF-13819](#).
- A composite index on some of the columns in SpacePermissions table is described in [CONF-14488](#).

Tuning the Database Connection Pool

This is described in the knowledge base article [Confluence Slows and Times out During Periods of High Load due to DB Connection Pool](#).

Configure a Database Query Timeout

If a database is getting overloaded, you can prevent it from crashing Confluence by [Configuring a Database Query Timeout](#).

Related Articles

[Troubleshooting Database Issues](#).

Creating a Lowercase Page Title Index

Diagnosis

Confluence sometimes has performance problems retrieving pages by title because the query uses the lower() function. For example, the query looks something like this:

```
select * from CONTENT where lower(TITLE) = :title and SPACEID = :spaceid
```

Database profiling might show a query like the following taking a long time to execute (emphasis added):

```
select ... from CONTENT page0_, SPACES space1_
where page0_.CONTENTTYPE='PAGE'
and ((lower(space1_.SPACEKEY)= @P0 and page0_.SPACEID=space1_.SPACEID)
```

```
and(lower(page0_.TITLE)= @P1)
and(page0_.PREVVER is null )and(page0_.CONTENT_STATUS='current' ))
```

Typically, databases don't use indexes when you use a function in a where clause; they do a table scan instead. This makes the performance of this query not ideal ([CONF-11577](#)).

Generic solution

On many databases (e.g. Oracle, PostgreSQL, DB2 for z/OS), it is possible to create the index using the normal "create index" syntax, just using the function instead of the column name.

```
create index CONFTITLE_LOWER on CONTENT(lower(TITLE));
```

Sources:

- <http://www.postgresql.org/docs/current/static/sql-createindex.html>
- <http://asktom.oracle.com/tkyte/article1/>

SQL Server

On SQL Server, you can add a computed column to the database table and then add an index on this column.

```
alter table CONTENT add TITLE_LOWER as lower(TITLE);
create index CONFTITLE_LOWER on CONTENT(TITLE_LOWER);
```

Sources:

- [http://msdn.microsoft.com/en-us/library/aa258260\(SQL.80\).aspx](http://msdn.microsoft.com/en-us/library/aa258260(SQL.80).aspx)
- <http://blogs.msdn.com/psssql/archive/2009/03/09/how-to-use-computed-columns-to-improve-query-performance.aspx>

MySQL

It is not currently possible to create a lowercase index on MySQL. Confluence 3.0 includes some caching improvements which should alleviate this performance problem on this database.

Source:

- <http://dev.mysql.com/doc/refman/5.1/en/create-index.html>

Workaround for MySQL databases, using a case-insensitive collation:

Please check whether your MySQL database has been set to use case-sensitive or case-insensitive collation. The queries to check whether your database is set to case-insensitive collation are:

```
show full columns from content where field = 'title';
show full columns from spaces where field = 'spacekey';
```

If the **collation_name** is returned as **<encoding>_ci**, the **ci** indicates case-insensitive collation.

If the database has been set to use case-insensitive collation, you can try removing **lower** from the following queries, in your

ContentEntityObject.hbm.xml file residing in your

<Confluence-Install>/confluence/WEB-INF/lib/confluence-2.x.x.jar/com/atlassian/confluence/core/:

```
<query name="confluence.page_findLatestBySpaceKeyTitle"><![CDATA[
  from Page page
  where lower(page.space.key) = :spaceKey and
         lower(page.title) = :pageTitle and
         page.originalVersion is null and
         page.contentStatus = 'current'
]]></query>

<query name="confluence.page_findLatestBySpaceKeyTitleOptimisedForComments"><![CDATA[
  from Page page
  left join fetch page.comments as theComments
  left join fetch theComments.children
  where lower(page.space.key) = :spaceKey and
         lower(page.title) = :pageTitle and
         page.originalVersion is null and
         page.contentStatus = 'current'
]]></query>
```

DB2 for Linux or Windows

DB2 supports [indexes on generated columns](#) which are used for queries with a matching predicate. You can implement it like this:

```
ALTER TABLE CONTENT ADD COLUMN TITLE_LOWER GENERATED ALWAYS AS (LOWER(TITLE));
CREATE INDEX CONFTITLE_LOWER ON CONTENT(TITLE_LOWER)
```

Related pages

- [Improving Database Performance](#)
- [CONF-10030: Queries that use 'lower' do not use index because of case sensitivity](#)

Webserver Configuration

- [Apache and Apache Connector Tips](#)
- [Configure Web Proxy Support for Confluence](#)
- [Running Confluence behind Apache](#)
 - [General Apache Configuration Notes](#)
 - [Using Apache with mod_jk](#)
 - [Using Apache with mod_proxy](#)
 - [Using Apache with virtual hosts and mod_proxy](#)
 - [Using mod_rewrite to Modify Confluence URLs](#)

Apache and Apache Connector Tips

The speed of downloading attachments is extremely slow. We are experiencing the following speeds

Large file served directly through Apache: 15000 KB/sec
 Large file served directly from Tomcat HTTP connector: 14500 KB/sec
 Large file served from Confluence (using Apache/mod_jk/Tomcat): 84 KB/sec

You can see that the file served from Confluence is ~176 times slower!

Solution

We upgraded mod_jk from version 1.2.8 to 1.2.10 and the download speed improved significantly to ~12000 KB/sec.

Configure Web Proxy Support for Confluence

Some of Confluence's macros, such as {rss} and {jiraissues} need to make web requests to remote servers in order to retrieve data. If Confluence is deployed within a data-centre or DMZ, it may not be able to access the Internet directly to make these requests. If you find that the {rss} macro does not work, ask your network administrator if Confluence needs to access the Internet through a web proxy.

Configuring an outbound HTTP proxy in Confluence

Proxy support is configured by passing certain [system properties](#) to the Java Virtual Machine on startup. These properties follow the [conventions defined by Oracle](#):

- `http.proxyHost`
- `http.proxyPort` (default: 80)
- `http.nonProxyHosts` (default: <none>)

At a minimum, you need to define `http.proxyHost` to configure an HTTP proxy. System property configuration is described on the [Configuring System Properties](#) page.

Properties `http.proxyHost` and `http.proxyPort` indicate the proxy server and port that the http protocol handler will use.

```
-Dhttp.proxyHost=proxy.example.org -Dhttp.proxyPort=8080
```

Property `http.nonProxyHosts` indicates the hosts which should be connected to directly and not through the proxy server. The value can be a list of hosts, each separated by a pipe character `|`. In addition, a wildcard character (asterisk) `*` can be used for matching. For example:

```
-Dhttp.nonProxyHosts=*.foo.com|localhost
```

Note: You may need to escape the pipe character `|` in some command-line environments.

If the `http.nonProxyHosts` property is not configured, all web requests will be sent to the proxy.



Please note that any command line parameters set are visible from the process list, and thus anyone who has the appropriate access to view the process list will see the proxy information in the clear.

To avoid this, you can set these properties in the `catalina.properties` file, located in `confluence-install/conf/`

Add this to the end of the file:

```
http.proxyHost=yourProxyURL
http.proxyPort=yourProxyPort
http.proxyUser=yourUserName
http.proxyPassword=yourPassword
```

Configuring HTTP proxy authentication

Proxy authentication is also configured by providing [system properties](#) to Java in your application server's configuration file. Specifically, the following two properties:

- `http.proxyUser` – username
- `http.proxyPassword` – secret

Authentication has a few more options in Confluence 2.10 and later, as documented below.

HTTP proxy (Microsoft ISA) NTLM authentication (2.10 and later)

Confluence 2.10 and later supports NTLM authentication for outbound HTTP proxies when Confluence is running on a Windows server.

To clarify, this means the `{rss}` and `{jiraissues}` macro will be able to contact external websites if requests have to go through a proxy that requires Windows authentication. This support is not related to logging in Confluence users automatically with NTLM, for which there is a [user-contributed authenticator](#) available.

To configure NTLM authentication for your HTTP proxy, you need to define a domain [system property](#), `http.auth.ntlm.domain`, in addition to the properties for host, port and username mentioned above:

```
-Dhttp.auth.ntlm.domain=MYDOMAIN
```

Configuring authentication order (2.10 and later)

Sometimes multiple authentication mechanisms are provided by an HTTP proxy. If you have proxy authentication failure messages, you should first check your username and password, then you can check for this problem by examining the HTTP headers in the proxy failure with a packet sniffer on the Confluence server. (Describing this is outside the scope of this document.)

To set the order for multiple authentication methods, you can set the [system property](#) `http.proxyAuth` to a comma-separated list of authentication methods. The available methods are: `ntlm`, `digest` and `basic`; this is also the default order for these methods.

For example, to attempt Basic authentication before NTLM authentication, and avoid Digest authentication entirely, you can set the `http.proxyAuth` property to this value:

```
-Dhttp.proxyAuth=basic,ntlm
```

Troubleshooting

1. There's a diagnostic jsp file in [CONF-9719](#) for assessing the connection parameters.
2. 'Status Code [407]' errors are described in [APR-160](#).
3. Autoproxies are not supported yet. See [CONF-16941](#).

Running Confluence behind Apache



The content on this page relates to platforms which are not supported for Confluence. Consequently, Atlassian **can not guarantee providing any support for the steps described on this page**. Please be aware that this material is provided for your information only and that you use it at your own risk.

Introduction

Running Confluence behind a web server should be done for performance reasons in high-load environments. In general, web server caching and thread management is far superior to that provided by your application server's HTTP interface.

To run Confluence behind the [Apache httpd web server](#), there are two main configuration options: `mod_jk` or `mod_proxy`.

Connection type	Features
<code>mod_proxy</code> (also known as reverse proxy)	<ul style="list-style-type: none"> • recommended connection method • simple HTTP proxy to application server • works with all application servers • if application paths are consistent, there is minimal load on the web server
<code>mod_jk</code> (also known as AJP)	<ul style="list-style-type: none"> • uses the AJP binary protocol • provides failover (and load balancing, which Confluence supports only with a clustered license) • only works with some application servers (typically Tomcat) • if application paths are consistent, there is some load on the web server to translate requests to AJP

Features common to both `mod_proxy` and `mod_jk`

- application paths must be consistent to avoid complex and slow URL rewriting
- works with name-based virtual hosting, both on web server and app server
- web server keeps a pool of connections to application server

Mod_proxy documentation

- [Using Apache with mod_proxy](#) is the main documentation for this configuration.
- If you want to set up the common configuration of JIRA and Confluence virtual hosts, you can use [Apache's virtual hosts](#) with separate application servers.

Mod_jk documentation

- [Using Apache with mod_jk](#) is the main documentation for this configuration.
- You can follow a similar method to the `mod_proxy` documentation above for setting up virtual hosts in Apache and Tomcat, if required.

Mod_jk2 not supported

The misleadingly-named `mod_jk2` is an older method of connecting to Tomcat from Apache. Since `mod_jk2` is [no longer supported by the Apache Foundation](#), we do not support this configuration, and are not updating our [mod_jk2 documentation](#). `mod_jk2` also has unresolved problems with Unicode URLs; you need to use either `mod_proxy` or `mod_jk` for international characters to work correctly in Confluence.

Other related documentation

- [Apache and Apache Connector Tips](#)
- [Using the \(older\) mod_jk2 connector](#)
- [Configuring Tomcat's URI encoding](#)
- [Running Confluence Over SSL or HTTPS](#)

General Apache Configuration Notes

On this page:

- [Prefer Apache mod_deflate to Confluence's built-in gzip implementation](#)
- [Ensure keepalive is enabled](#)
- [Enable keepalive for recent MSIE user agents](#)

Prefer Apache mod_deflate to Confluence's built-in gzip implementation

1. Disable gzip in confluence. See [Compressing an HTTP Response within Confluence](#).
2. Enable gzip compression in Apache. For RedHat distributions this can be achieved by adding the following lines:

```
AddOutputFilterByType DEFLATE text/html text/plain text/xml text/css
application/x-javascript

# ensure sensible defaults
DeflateBufferSize 8192
DeflateCompressionLevel 4
DeflateMemLevel 9
DeflateWindowSize 15
```

Ensure keepalive is enabled

```
KeepAlive On
```

Enable keepalive for recent MSIE user agents

The standard Apache SSL configuration is very conservative when it comes to MSIE and SSL. By default all keepalives are disabled when using HTTPS with MSIE. While MSIE will always be *special*, the issues with SSL and MSIE have been solved since [Service Pack 2 for Windows XP](#), released over 4 years go. For anyone using an XP machine SP2 or above, it is safe to allow keepalive for MSIE 6 and above.

Remove the following lines:

```
SetEnvIf User-Agent ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
```

Add these in their place:

```
BrowserMatch "MSIE [1-5]" nokeepalive ssl-unclean-shutdown downgrade-1.0 force-response-1.0
BrowserMatch "MSIE [6-9]" ssl-unclean-shutdown
```

RELATED TOPICS

[Running Confluence behind Apache](#)
[Apache and Apache Connector Tips](#)
[Configuring Tomcat's URI encoding](#)
[Running Confluence Over SSL or HTTPS](#)

Using Apache with mod_jk



The content on this page relates to platforms which are not supported for Confluence. Consequently, Atlassian **can not guarantee providing any support for the steps described on this page**. Please be aware that this material is provided for your information only and that you use it at your own risk.

Introduction

The Apache web server is often used in front of an application server to improve performance in high-load environments. Mod_jk allows request forwarding to an application via a protocol called AJP. Configuration of this involves enabling mod_jk in Apache, configuring a AJP connector in your application server, and directing Apache to forward certain paths to the application server via mod_jk.

Mod_jk is sometimes preferred to [mod_proxy](#) because AJP is a binary protocol, and because some site administrators are more familiar with it than with mod_proxy..

The scope of this documentation is limited to configuring the AJP connector in Tomcat 5.x. Other application servers may support AJP connectors; please consult your application server documentation for instructions on how to configure it.

The configuration below assumes your Confluence instance is accessible on the same path on the application server and the web server. For example:

Externally accessible (web server) URL	http://www.example.com/confluence/
Application server URL (HTTP)	http://app-server.internal.example.com:8090/confluence/

The AJP connection of the application server is set to: app-server.internal.example.com:8009.

Configuring mod_jk in Apache

The standard distribution of Apache does not include mod_jk. You need to download it from the [JK homepage](#) and put the mod_jk.so file in your Apache modules directory.

Next, add the following in **httpd.conf** directly or included from another file:

```
# Put this after the other LoadModule directives
LoadModule jk_module modules/mod_jk.so

# Put this in the main section of your configuration (or desired virtual host, if using Apache
virtual hosts)
JkWorkersFile conf/workers.properties
JkLogFile logs/mod_jk.log
JkLogLevel info

JkMount /confluence worker1
JkMount /confluence/* worker1
```

Configuring workers.properties

Create a new file called 'workers.properties', and put it in your Apache conf directory. (The path for workers.properties was one of the configuration settings above.)

```
worker.list=worker1

worker.worker1.host=app-server.internal.example.com
worker.worker1.port=8009
worker.worker1.type=ajp13
```

Tomcat 5.x configuration

In Tomcat 5, the AJP connector is enabled by default on port 8009. An absolutely minimal Tomcat server.xml is below for comparison. The relevant line is the Connector with port 8009 – make sure this is uncommented in your server.xml.

```
<Server port="8000" shutdown="SHUTDOWN">
  <Service name="Catalina">

    <!-- Define a HTTP/1.1 Connector on port 8090 -->
    <Connector port="8090" />

    <!-- Define an AJP 1.3 Connector on port 8009 -->
    <Connector port="8009" protocol="AJP/1.3" />

    <Engine name="Catalina" defaultHost="localhost">
      <Host name="localhost" appBase="webapps">
        <Context path="/confluence" docBase="/opt/webapps/confluence-2.2/confluence" />
        <Logger className="org.apache.catalina.logger.FileLogger" />
      </Host>
    </Engine>
  </Service>
</Server>
```

Points to note:

- the Connector on port 8009 has protocol of "AJP/1.3". This is critical.
- the Context path of the Confluence application is "/confluence". This must match the path used to access Confluence on the web server.
- we recommend keeping your application Contexts outside the server.xml in Tomcat 5.x. The above example includes them for demonstration only.

Improving the performance of the mod_jk connector

The most important setting in high-load environments is the number of processor threads used by the Tomcat AJP connector. By default, this is 200, but you should increase it to match Apache's maxThreads setting (256 by default):

```
<Connector port="8009" minSpareThreads="5" maxThreads="256" protocol="AJP/1.3" />
```

All the configuration parameters for the AJP connector are covered in [the Tomcat documentation](#).

Ensuring UTF-8 compatibility

If you have problems downloading attachments with non-ASCII characters in the filename, add the following to your Apache configuration:

```
JkOptions +ForwardURICompatUnparsed
```

And specify UTF-8 as the URIEncoding in the AJP connector configuration:

```
<Connector port="8009" protocol="AJP/1.3" URIEncoding="UTF-8" />
```

These settings are discussed further on [Configuring Tomcat's URI encoding](#).

More information

The [Tomcat JK website](#) has complete documentation on workers.properties and Apache configuration. You can also find information there on how to use mod_jk with IIS.

Alternatives

If you're not happy with mod_jk, or find it too difficult to configure, you can:

- use [mod_proxy](#), which works with any application server, and together with mod_proxy_html allows complex URL rewriting to deal with different application paths on the web server and the application server.

Using Apache with mod_proxy

This page describes how to integrate Confluence into an Apache website using mod_proxy.



The content on this page relates to platforms which are not supported for Confluence. Consequently, Atlassian **can not guarantee providing any support for the steps described on this page**. Please be aware that this material is provided for your information only and that you use it at your own risk.

On this page:

- [Simple configuration](#)
 - [Set the context path](#)
 - [Configure mod_proxy](#)
 - [Set the URL for redirection](#)
- [Complex configuration](#)
- [Adding SSL](#)
 - [More information](#)
 - [Alternatives](#)

There are some common situations where you might do this:

- You have an existing Apache-based website, and want to add Confluence to the mix (for example, <http://www.example.com/confluence>).
- You have two or more Java applications, each running in their own application server on different ports, for example, <http://localhost:8090/confluence> and <http://localhost:8080/jira>. By setting up Apache with mod_proxy, you can have both available on the regular HTTP port (80) – for example, at <http://www.example.com/confluence> and <http://www.example.com/jira>. This allows each application to be restarted, managed and debugged separately.

This page describes how to configure mod_proxy. We describe two options:

- If you want a URL like <http://www.example.com/confluence/>, go to the [simple configuration](#).
- If you want a URL like <http://confluence.example.com/>, go to the [complex configuration](#).

Simple configuration

Set the context path

First, set your Confluence application path (the part after hostname and port) correctly. Say you want Confluence available at <http://www.example.com/confluence/>, and you currently have it running at <http://localhost:8090/>. The first step is to get Confluence available at <http://localhost:8090/confluence/>.

To do this in Tomcat (bundled with Confluence), edit `conf/server.xml`, locate the "Context" definition:

```
<Context path="" docBase="../confluence" debug="0" reloadable="true">
```

and change it to:

```
<Context path="/confluence" docBase="../confluence" debug="0" reloadable="true">
```

Then restart Confluence, and ensure you can access it at <http://localhost:8090/confluence/>

Configure mod_proxy

Now enable mod_proxy in Apache, and proxy requests to the application server by adding the example below to your Apache `httpd.conf` (note: the files may be different on your system; the [JIRA docs](#) describe the process for Ubuntu/Debian layout):

```
# Put this after the other LoadModule directives
LoadModule proxy_module /usr/lib/apache2/modules/mod_proxy.so
LoadModule proxy_http_module /usr/lib/apache2/modules/mod_proxy_http.so

# Put this in the main section of your configuration (or desired virtual host, if using Apache
virtual hosts)
ProxyRequests Off
ProxyPreserveHost On

<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

ProxyPass /confluence http://localhost:8090/confluence
ProxyPassReverse /confluence http://localhost:8090/confluence
<Location /confluence>
    Order allow,deny
    Allow from all
</Location>
```

▼ Note to Windows Users

It is recommended that you specify the absolute path to the `mod_proxy.so` and `mod_proxy_http.so` files.

Set the URL for redirection

You will need to modify the `server.xml` file in your tomcat's `conf` directory and set the URL for redirection.

Locate this code segment

```
<Connector port="8090" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" redirectPort="8443" acceptCount="100"
    connectionTimeout="20000" disableUploadTimeout="true" />
```

And append the following segment:

```
<Connector port="8090" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" redirectPort="8443" acceptCount="100"
    connectionTimeout="20000" disableUploadTimeout="true"
    proxyName="www.example.com" proxyPort="80" />
```

Replace **www.example.com** with the URL you wish to be redirected to.



If this isn't working for you, try adding a scheme attribute to your Connector tag: **scheme="https"**.

Complex configuration

Complex configuration involves using the `mod_proxy_html` filter to modify the proxied content en-route. This is required if the Confluence path differs between Apache and the application server. For example:

Externally accessible (Apache) URL	http://confluence.example.com/
Application server URL	http://app-server.internal.example.com:8090/confluence/

Notice that the application path in the URL is different in each. On Apache, the path is `/`, and on the application server the path is `/confluence`.



For this configuration, you need to install the `mod_proxy_html` module, which is not included in the standard Apache distribution.

Alternative solutions are discussed below.

```
# Put this after the other LoadModule directives
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_html_module modules/mod_proxy_html.so

<VirtualHost *>
    ServerName confluence.example.com

    # Put this in the main section of your configuration (or desired virtual host, if using
    Apache virtual hosts)
    ProxyRequests Off
    ProxyPreserveHost On

    <Proxy *>
        Order deny,allow
        Allow from all
    </Proxy>

    ProxyPass / http://app-server.internal.example.com:8090/confluence
    ProxyPassReverse / http://app-server.internal.example.com:8090/confluence

    ProxyHTMLURLMap / /confluence/

    <Location />
        Order allow,deny
        Allow from all
    </Location>
</VirtualHost>
```

The ProxyHTMLURLMap configuration can become more complex if you have multiple applications running under this configuration. The mapping should also be placed in a Location block if the web server URL is a subdirectory and not on a virtual host. The [Apache Week tutorial](#) has more information how to do this.

Adding SSL

If you're running Apache in front of Tomcat, it's a good idea to terminate your SSL configuration at Apache, then forward the requests to Tomcat over HTTP. You can set up Apache to terminate the SSL connection and use the ProxyPass and ProxyPassReverse directives to pass the connection through to Tomcat (or the appropriate application server) which is running Confluence.

1. Create a new SSL host by creating a virtual host on 443
2. The standard http connection on apache could be used to redirect to https if you want or it could just be firewalled.
3. Within the VirtualHost definition:
 - a. define the SSL options (SSLEngine and SSLCertificateFile)
 - b. define the ProxyPass and ProxyPassReverse directives to pass through to Tomcat.

Because of how the ProxyPass and ProxyPassReverse directives work, you should not need to modify the tomcat installation at all.

Most of the relevant Apache Config:

```
Listen 443

NameVirtualHost *:443
<VirtualHost *:443>
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/apache.pem
    ProxyPass / http://localhost:8090/
    ProxyPassReverse / http://localhost:8090/
</VirtualHost>
```

More information

- The [mod_proxy_html site](#) has documentation and examples on the use of this module in the complex configuration.

- [Apache Week has a tutorial](#) that deals with a complex situation involving two applications and ProxyHTMLURLMap.
- [Using Apache with virtual hosts and mod_proxy](#) shows how to configure the special case where you want JIRA and Confluence running on separate application servers on virtual host subdomains.

Alternatives

If Tomcat is your application server, you have two options:

- use [mod_jk](#) to send the requests to Tomcat
- use [Tomcat's virtual hosts](#) to make your Confluence application directory the same on the app server and the web server, removing the need for the URL mapping.

If your application server has an AJP connector, you can:

- use [mod_jk](#) to send the requests to your application server.

Using Apache with virtual hosts and mod_proxy



The content on this page relates to platforms which are not supported for Confluence. Consequently, Atlassian **can not guarantee providing any support for the steps described on this page**. Please be aware that this material is provided for your information only and that you use it at your own risk.

Introduction

The Apache web server is often used in front of an application server to improve performance in high-load environments. Mod_proxy simply redirects requests for certain URLs to another web server, so it typically requires no additional configuration on the application server.

This page documents a very common configuration request: configuring JIRA and Confluence on two Apache virtual hosts, running on different application servers. This is just a special case of [mod_proxy configuration](#).

You can use virtual hosts in your application server if you want to run JIRA and Confluence on the same application server. There is a [sample configuration for Tomcat](#) you can use after configuring Apache.

Apache configuration

For this configuration to work properly, the application paths must be the same on both the application servers and the web server. For both JIRA and Confluence below, this is `/`.

JIRA external URL	http://jira.example.com/
JIRA application server URL	http://jira-app-server.internal.example.com:8080/
Confluence external URL	http://confluence.example.com/
Confluence application server URL	http://confluence-app-server.internal.example.com:8090/

Add the following to your Apache httpd.conf:


```
# Put this after the other LoadModule directives
LoadModule proxy_module /usr/lib/apache2/modules/mod_proxy.so
LoadModule proxy_http_module /usr/lib/apache2/modules/mod_proxy_http.so

# Put this with your other VirtualHosts, or at the bottom of the file
NameVirtualHost *
<VirtualHost *>
    ServerName confluence.example.com

    ProxyRequests Off
    <Proxy *>
        Order deny,allow
        Allow from all
    </Proxy>

    ProxyPass / http://confluence-app-server.internal.example.com:8090/
    ProxyPassReverse / http://confluence-app-server.internal.example.com:8090/
    <Location />
        Order allow,deny
        Allow from all
    </Location>
</VirtualHost>
<VirtualHost *>
    ServerName jira.example.com

    ProxyRequests Off
    <Proxy *>
        Order deny,allow
        Allow from all
    </Proxy>

    ProxyPass / http://jira-app-server.internal.example.com:8080/
    ProxyPassReverse / http://jira-app-server.internal.example.com:8080/
    <Location />
        Order allow,deny
        Allow from all
    </Location>
</VirtualHost>
```

Points to note:

- ProxyPass and ProxyPassReverse directives send traffic from the web server to your application server.
- The application path is the same on the application server and on the web server (both are /).
- Because the above configuration uses [name-based virtual hosting](#), you must configure your DNS server to point both names (jira.example.com, confluence.example.com) to your web server.

More information

For different ways to configure mod_proxy, see [Using Apache with mod_proxy](#).

If you use Tomcat, [mod_jk](#) provides a different way of connecting Apache via AJP. You can also use the above configuration with just one application server if you use Tomcat's virtual hosts.

Using mod_rewrite to Modify Confluence URLs



Confluence requires URL rewriting for proper functionality. If Confluence is accessible via different domain names *without* URL rewriting, you will experience an array of problems. See [Various Issues Caused when Server Base URL Does Not Match the URL Used to Access Confluence](#).

In some cases, you may wish to access Confluence from different domains.

- <http://wiki> (from an internal network)
- <http://wiki.domain.com> (the externally visible domain)

To configure Confluence over multiple domains,

1. Add a DNS entry mapping <http://wiki> to the externally visible ip address of the Confluence server.
2. Configure Confluence's [server base url](#) to <http://wiki.domain.com>.
3. Add Apache HTTP proxy, using the instructions from [Running Confluence behind Apache](#).
4. Add the `mod_rewrite` module, to change the URL.

Start Confluence Automatically on System Startup

Confluence can be configured to start automatically on system startup, allowing it to recover automatically after a reboot.

Page: [Start Confluence Automatically on Windows as a Service](#)

Page: [Start Confluence Automatically on Linux](#)

Start Confluence Automatically on Linux

On Linux/Solaris, the best practice is to install, configure and run each service (including Confluence) as a dedicated user with only the permissions they require.

To install, configure and run Confluence automatically on Linux/Solaris:

1. Create a `confluence` user for instance, using the following command:

```
sudo useradd --create-home -c "Confluence role account" confluence
```

2. Create a directory to install Confluence into:

```
sudo mkdir /usr/local/confluence
sudo chown confluence: /usr/local/confluence
```

3. Log in as the `confluence` user to install Confluence:

```
sudo su - confluence
cd /usr/local/confluence/
tar zxvf /tmp/confluence-3.0.1-std.tar.gz
ln -s confluence-3.0.1-std/ current
```

4. Edit `<<CONFLUENCE_INSTALL_DIRECTORY>>/confluence/WEB-INF/classes/confluence-init.properties` file, and set `confluence.home=/usr/local/confluence/<Confluence_Data_Home>` (ensure you have removed the comment '#')
5. Then back as root, create the file `/etc/init.d/confluence` (code shown below), which will be responsible for starting up Confluence after a reboot (or when manually invoked).



If you are running Ubuntu Jaunty (or later) do not perform this step. Please use the [instructions](#) further down this page.

```
#!/bin/sh -e
# Confluence startup script
#chkconfig: 2345 80 05
#description: Confluence

# Define some variables
# Name of app ( JIRA, Confluence, etc )
APP=confluence
# Name of the user to run as
USER=confluence
# Location of application's bin directory
CATALINA_HOME=/usr/local/confluence/current
# Location of Java JDK
export JAVA_HOME=/usr/lib/jvm/java-6-sun

case "$1" in
# Start command
start)
    echo "Starting $APP"
    /bin/su -m $USER -c "$CATALINA_HOME/bin/startup.sh &> /dev/null"
    ;;
# Stop command
stop)
    echo "Stopping $APP"
    /bin/su -m $USER -c "$CATALINA_HOME/bin/shutdown.sh &> /dev/null"
    echo "$APP stopped successfully"
    ;;
# Restart command
restart)
    $0 stop
    sleep 5
    $0 start
    ;;
*)
    echo "Usage: /etc/init.d/$APP {start|restart|stop}"
    exit 1
    ;;
esac

exit 0
```


6. Make this file executable:

```
sudo chmod +x /etc/init.d/confluence
```

7. Set this file to run at the appropriate runlevel. For example, use `sudo chkconfig --add confluence` on Redhat-based systems, `sudo update-rc.d confluence defaults` or `rcconf` on Debian-based systems.
8. You should now be able to start Confluence with the init script. A successful startup output typically looks like this:

```
$ sudo /etc/init.d/confluence start
Starting Confluence:
If you encounter issues starting up Confluence, please see the Installation guide at
http://confluence.atlassian.com/display/DOC/Confluence+Installation+Guide
Using CATALINA_BASE:   /usr/local/confluence/current
Using CATALINA_HOME:   /usr/local/confluence/current
Using CATALINA_TMPDIR: /usr/local/confluence/current/temp
Using JRE_HOME:        /usr/lib/jvm/java-1.5.0-sun
done.
```

You should then see this running at `http://<server>:8090/`

 The port for this will be whatever is defined in your Confluence `server.xml` file.

Adding Confluence as a service for Ubuntu Jaunty (or later)

To continue configuring Confluence to start automatically as a service on Ubuntu Jaunty (or later):

1. After **logging in** as the **confluence** user to install Confluence, create start and stop scripts in `/usr/local/confluence`:

Example startscript:

```
#!/bin/bash
export JAVA_HOME=/usr/lib/jvm/java-6-sun-1.6.0.16/
export JDK_HOME=/usr/lib/jvm/java-6-sun-1.6.0.16/
cd /usr/local/confluence/current/bin
./startup.sh
```

Example stopscript:

```
#!/bin/bash
export JAVA_HOME=/usr/lib/jvm/java-6-sun-1.6.0.16/
export JDK_HOME=/usr/lib/jvm/java-6-sun-1.6.0.16/
cd /usr/local/confluence/current/bin
./shutdown.sh
```

2. Make both of these scripts executable. For example, by issuing the command: `sudo chmod a+x /usr/local/confluence/start /usr/local/confluence/stop`.
3. Karmic and later: Create two text files in `/etc/init/` called `confluence-up.conf` and `confluence-down.conf`:

confluence-up:

```
start on runlevel [2345]

script

date >> /tmp/confluence-startup.out
exec sudo -u confluence /usr/local/confluence/start >> /tmp/confluence-startup.out 2>&1

end script
```

confluence-down:

```
start on runlevel [16]

expect fork
respawn

exec sudo -u confluence /usr/local/confluence/stop >> /tmp/confluence-shutdown.out 2>&1
```

... and make them readable to all users:

```
sudo chmod a+r /etc/init/confluence-up.conf /etc/init/confluence-down.conf
```

1. Jaunty, Intrepid: Create two text files in `/etc/event.d/` called `confluence-up` and `confluence-down`:

confluence-up:

```
start on runlevel 2
start on runlevel 3
start on runlevel 4
start on runlevel 5

exec sudo -u confluence /usr/local/confluence/start >> /tmp/confluence-startup.out 2>&1
```

confluence-down:

```
start on runlevel 1
start on runlevel 6

exec sudo -u confluence /usr/local/confluence/stop >> /tmp/confluence-shutdown.out 2>&1
```

... and make them readable to all users:

```
sudo chmod a+r /etc/event.d/confluence-up /etc/event.d/confluence-down
```

RELATED TOPICS

[Start Confluence Automatically on System Startup](#)

Start Confluence Automatically on Windows as a Service

For long-term use, we recommend that you configure Confluence to start automatically when the operating system restarts. For Windows servers, this means configuring Confluence to run as a Windows service.

There are two ways to install the Confluence distribution as a service: using the [Confluence installer](#) or manually as described below.

On this page:

- [Reasons for Starting Confluence as a Service](#)
- [Changing the User Running the Service](#)
- [Manually Installing the Confluence Distribution as a Service](#)
- [Managing Confluence as a Service](#)
- [Upgrading Confluence](#)
- [Troubleshooting Confluence while Running as a Windows Service](#)
- [Requesting Support](#)



Problem with 64-bit Windows

If you are running 64-bit Windows, please note that Apache Tomcat cannot run as a Windows service if you are using a 64-bit JDK. **Please ensure that you are using a 32-bit JDK.** Refer to our [knowledge base article](#) for more information.

Reasons for Starting Confluence as a Service

Installation as a Windows service offers these advantages:

- Reduced risk of shutting down Confluence by accident (If you start Confluence manually, a console window opens and there is a risk of someone accidentally shutting down Confluence by closing the window).
- Automated Confluence recovery after server restart.
- Improved troubleshooting through logging server output to file.

You can read more about Windows services in the [Microsoft Developer Network](#).

Changing the User Running the Service

If you wish to run the service as a non-administrator user for security, or if you are using network drives for backups, attachments or indexes, you can run the service as another user. To change users, open the Apache Tomcat Confluence properties, go to the 'Log On' tab and enter the required username and password. Go to your Windows Control Panel -> User Accounts and confirm that the user has write permissions for the %CATALINA_HOME%, index and database directories. Note that any network drives must be specified by UNC and not letter mappings (eg. \\backupserver\confluence not z:\confluence).

For more detail, see [Creating a Dedicated User Account on the Operating System to Run Confluence](#).

Manually Installing the Confluence Distribution as a Service

From your Windows-based server:

1. Open a command prompt in the <CONFLUENCE-INSTALL>/bin directory.
2. Confirm that the JAVA_HOME variable is set to the JDK base directory with the command:

```
echo %JAVA_HOME%
```

Note that any directory in the path with spaces (eg. C:\Program Files) must be converted to its eight-character equivalent (e.g. C:\Progra~1).

3. If you are installing Confluence on a Windows 2008 server, be sure to run the command prompt using 'run as administrator'. (Otherwise running 'service.bat', as described in the next step, will fail.)
4. Use the following command to install the service with default settings:

```
service.bat install Confluence
```

5. Now, to have the service start automatically when the server starts, run:

```
tomcat6 //US//Confluence --Startup auto
```


6. If you have a less than a 512 megabytes of memory, skip this step. For users with large Confluence installations, you can increase the maximum memory Confluence can use. (The default is 256MB). For example, you can set the maximum memory to 512 megs using:

```
tomcat6 //US//Confluence --JvMx 512
```

7. If you do not have any JVM parameters that you pass to your distribution of Confluence, you can skip this step. If you do, add them to the service using:

```
tomcat6 //US//Confluence ++JvmOptions="-Djust.an.example=True"
```

8. For further configuration options, please refer to the [Tomcat Windows Service How-To](#) guide
9. Go to your Windows Control Panel -> Administrative Tools -> Services -> Apache Tomcat Confluence and right-click on Properties to verify the settings are correct.

10.  Confluence is now installed as a service, but will not automatically start up until the next server reboot
10. Start the Confluence service with the command:

```
net start Confluence
```

Managing Confluence as a Service

You can manage the Confluence service from the command prompt.

- Stop Confluence with:

```
net stop Confluence
```

- Uninstall the Confluence service with:

```
service.bat remove Confluence
```

Upgrading Confluence

After upgrading Confluence, you can either uninstall and reinstall the Windows service or change the StartPath parameter to your new folder. Refer to the [Tomcat documentation](#) for help.

Troubleshooting Confluence while Running as a Windows Service

- Check the Knowledge Base articles:

Page: [Unable to Install Service on Windows Vista](#)

Page: [Unable to Configure Confluence to Run as a Service on Tomcat 5](#)

Page: [Confluence Does Not Start Due to Windows Firewall](#)

Page: [Problems Installing Confluence as a Service on Windows 64bit](#)

Page: [Getting 'The image file tomcat6.exe is valid, but is for a machine type other than the current machine'](#)

Page: [Unable to Start Confluence Windows Service After Allocating JVM Memory](#)

- If none of the above solves your problem, please refer to the complete [list of known issues](#) in our Knowledge Base.
- When investigating memory issues or bugs, it may be useful to view information from Confluence's garbage collection. To turn on the verbose garbage collection, use the command:

```
tomcat6 //US//Confluence
++JvmOptions="-Xloggc:<CONFLUENCE-INSTALL>\logs\atlassian-gc.log"
```

- The Confluence 2.9 installer does not work when installed as service, due to a missing semi-colon in `service.bat`. Please refer to reported issue [CONF-12785](#).
- You can use a [Sysinternals](#) tool called **Procmon.exe** from the The Microsoft Windows Sysinternals Team, to check that the error occurred at the specific time when the Confluence service started. You need to match the time when Tomcat failed, as captured by this tool, against the time in the Windows Event Viewer.



Note

We do not recommend that you run this tool for too long as it may disrupt other Atlassian applications. Once you have captured the required information you will need to press **Ctrl + E** to stop capturing.

Requesting Support

If, after following the troubleshooting guide above, you still cannot make Confluence run as a Windows Service or if there is an error when setting the JVM configuration for the service, you can [create a support request](#).

Please provide the following information when creating your support request, because we will need it to assist you:

- Are you running a 32 bit or 64 bit Windows?
- Give us the result of running **java -version** from Windows command line console.
- A screen shot of your Windows Registry setting for Tomcat.
- If you have modified **service.bat**, please give us a copy of this file for review.
- What application server are you using? eg. Are you using the Confluence distribution?
- Your [atlassian-confluence.log](#) file.

RELATED TOPICS

[Start Confluence Automatically on System Startup](#)
[Fix Out of Memory Errors by Increasing Available Memory](#)

Confluence Data Model

On this page:

- [General Database Diagram](#)
- [Authentication](#)
 - [Atlassian-user](#)
 - [OpenSymphony](#)
- [Content](#)
- [Clustering](#)
 - [System information](#)
 - [Spaces](#)
- [Appearance](#)
- [Miscellaneous](#)



The Hibernate mapping files are the authoritative reference. These are the *.hbm.xml files which have been bundled into the main Confluence .jar file in recent releases.

This document is little more than the Confluence schema with added comments, but the priority was placed on making the information available.

General Database Diagram



Authentication

Atlassian-user

This is the "new" authentication system, which is more flexible and extensible than OpenSymphony.

```
Table "groups"
  Column |          Type          | Modifiers
-----+-----+-----
  id     | bigint                 | not null
  groupname | character varying(255) | not null
Indexes:
    "groups_pkey" PRIMARY KEY, btree (id)
```

```
Table "users"
  Column |          Type          | Modifiers
-----+-----+-----
  id     | bigint                 | not null
  name   | character varying(255) | not null
  password | character varying(255) |
  email  | character varying(255) |
  created | timestamp without time zone |
  fullname | character varying(255) |
Indexes:
    "users_pkey" PRIMARY KEY, btree (id)
    "users_name_key" UNIQUE, btree (name)
```

`local_members`: establishes many-to-many association between users and groups.

```
Table "local_members"
  Column | Type | Modifiers
-----+-----+-----
  userid | bigint | not null
  groupid | bigint | not null
Indexes:
    "local_members_pkey" PRIMARY KEY, btree (groupid, userid)
Foreign-key constraints:
    "fk6b8fb445117d5fda" FOREIGN KEY (groupid) REFERENCES groups(id)
    "fk6b8fb445ce2b3226" FOREIGN KEY (userid) REFERENCES users(id)
```

`external_entities`: Maps users from LDAP (or any other external authentication system) to IDs in Confluence DB


```
Table "external_entities"
  Column |          Type          | Modifiers
-----+-----+-----
  id     | bigint                 | not null
  name   | character varying(255) | 
  type   | character varying(255) | not null
Indexes:
    "external_entities_pkey" PRIMARY KEY, btree (id)
```

external_members: associates LDAP (or other external) users with local groups.

```
Table "external_members"
  Column | Type | Modifiers
-----+-----+-----
  extentityid | bigint | not null
  groupid     | bigint | not null
Indexes:
    "external_members_pkey" PRIMARY KEY, btree (groupid, extentityid)
Foreign-key constraints:
    "fk8c8d8a5117d5fda" FOREIGN KEY (groupid) REFERENCES groups(id)
    "fk8c8d8a5f25e5d5f" FOREIGN KEY (extentityid) REFERENCES external_entities(id)
```

OpenSymphony

The "old" authentication system, which was the default prior to 2.7.

```
Table "os_group"
  Column |          Type          | Modifiers
-----+-----+-----
  id     | bigint                 | not null
  groupname | character varying(255) | not null
Indexes:
    "os_group_pkey" PRIMARY KEY, btree (id)
    "os_group_groupname_key" UNIQUE, btree (groupname)
```

```
Table "os_user"
  Column |          Type          | Modifiers
-----+-----+-----
  id     | bigint                 | not null
  username | character varying(255) | not null
  passwd  | character varying(255) | 
Indexes:
    "os_user_pkey" PRIMARY KEY, btree (id)
    "os_user_username_key" UNIQUE, btree (username)
```

```
Table "os_user_group"
  Column | Type | Modifiers
-----+-----+-----
  group_id | bigint | not null
  user_id  | bigint | not null
Indexes:
    "os_user_group_pkey" PRIMARY KEY, btree (user_id, group_id)
Foreign-key constraints:
    "fk932472461e2e76db" FOREIGN KEY (group_id) REFERENCES os_group(id)
    "fk93247246f73aee0f" FOREIGN KEY (user_id) REFERENCES os_user(id)
```

Content

The actual information that users are storing and sharing.

attachmentdata: stores the binary data for attached files.

Only used when Confluence is configured to store attachments in the database; otherwise, attachments are stored in the local filesystem.

```
Table "attachmentdata"
  Column      |  Type  | Modifiers
-----+-----+-----
attachmentdataid | bigint | not null
attversion      | integer | not null
data            | bytea  |
attachmentid    | bigint |
Indexes:
    "attachmentdata_pkey" PRIMARY KEY, btree (attachmentdataid)
    "attch_data_idx" btree (attachmentid)
Foreign-key constraints:
    "fk9dc3e34d34a4917e" FOREIGN KEY (attachmentid) REFERENCES attachments(attachmentid)
```

attachments: metadata for attachments.

```
Table "attachments"
  Column      |  Type  | Modifiers
-----+-----+-----
attachmentid  | bigint | not null
title         | character varying(255) | not null
contenttype   | character varying(255) | not null
pageid        | bigint | not null
creator       | character varying(255) |
creationdate  | timestamp without time zone |
lastmodifier  | character varying(255) |
lastmoddate   | timestamp without time zone |
filesize      | bigint |
attachment_comment | character varying(255) |
attversion    | integer |
prevver       | bigint |
Indexes:
    "attachments_pkey" PRIMARY KEY, btree (attachmentid)
    "att_pageid_idx" btree (pageid)
    "att_prevver_idx" btree (prevver)
Foreign-key constraints:
    "fk54475f9017d4a070" FOREIGN KEY (prevver) REFERENCES attachments(attachmentid)
    "fk54475f908c38fba" FOREIGN KEY (pageid) REFERENCES content(contentid)
```

bodycontent: stores the actual content of Confluence pages. No versioning information or other metadata is stored here, though; that's all in the content table.

```
Table "bodycontent"
  Column      |  Type  | Modifiers
-----+-----+-----
bodycontentid | bigint | not null
body          | text   |
contentid     | bigint |
Indexes:
    "bodycontent_pkey" PRIMARY KEY, btree (bodycontentid)
    "body_content_idx" btree (contentid)
Foreign-key constraints:
    "fka898d4778dd41734" FOREIGN KEY (contentid) REFERENCES content(contentid)
```

content: a persistence table for the ContentEntityObject class of objects. The subclass is indicated by the contenttype column.

Table "content"		
Column	Type	Modifiers
contentid	bigint	not null
contenttype	character varying(255)	not null
title	character varying(255)	
version	integer	
creator	character varying(255)	
creationdate	timestamp without time zone	
lastmodifier	character varying(255)	
lastmoddate	timestamp without time zone	
versioncomment	text	
prevver	bigint	
content_status	character varying(255)	
spaceid	bigint	
parentid	bigint	
messageid	character varying(255)	
draftpageid	character varying(255)	
draftspacekey	character varying(255)	
drafttype	character varying(255)	
draftpageversion	integer	
pageid	bigint	
parentcommentid	bigint	
username	character varying(255)	

Indexes:

"content_pkey" PRIMARY KEY, btree (contentid)
 "c_draftpageid_idx" btree (draftpageid)
 "c_draftspacekey_idx" btree (draftspacekey)
 "c_drafttype_idx" btree (drafttype)
 "c_messageid_idx" btree (messageid)
 "c_parentcommid_idx" btree (parentcommentid)
 "c_parentid_idx" btree (parentid)
 "c_prevver_idx" btree (prevver)
 "c_spaceid_idx" btree (spaceid)
 "c_title_idx" btree (title)
 "c_username_idx" btree (username)

Foreign-key constraints:

"fk6382c05917d4a070" FOREIGN KEY (prevver) REFERENCES content(contentid)
 "fk6382c05974b18345" FOREIGN KEY (parentid) REFERENCES content(contentid)
 "fk6382c0598c38fbea" FOREIGN KEY (pageid) REFERENCES content(contentid)
 "fk6382c059b2dc6081" FOREIGN KEY (spaceid) REFERENCES spaces(spaceid)
 "fk6382c059b97e9230" FOREIGN KEY (parentcommentid) REFERENCES content(contentid)

content_label: Arbitrary text labels for content.

Table "content_label"		
Column	Type	Modifiers
id	bigint	not null
labelid	bigint	not null
contentid	bigint	not null
spacekey	character varying(255)	
owner	character varying(255)	
creationdate	timestamp without time zone	
lastmoddate	timestamp without time zone	

Indexes:

"content_label_pkey" PRIMARY KEY, btree (id)
 "cl_contentid_idx" btree (contentid)
 "cl_labelid_idx" btree (labelid)
 "cl_lastmoddate_idx" btree (lastmoddate)
 "cl_spacekey_idx" btree (spacekey)

Foreign-key constraints:

"fkf0e7436e27072aef" FOREIGN KEY (labelid) REFERENCES label(labelid)
 "fkf0e7436e8dd41734" FOREIGN KEY (contentid) REFERENCES content(contentid)

label: the other half of the content_label system.

```

Table "label"
  Column      |          Type          | Modifiers
-----+-----+-----
labelid      | bigint                 | not null
name         | character varying(255) |
owner        | character varying(255) |
namespace    | character varying(255) |
creationdate | timestamp without time zone |
lastmoddate  | timestamp without time zone |
Indexes:
    "label_pkey" PRIMARY KEY, btree (labelid)
    "l_name_idx" btree (name)
    "l_namespace_idx" btree (namespace)
    "l_owner_idx" btree ("owner")

```

`content_perm`: content-level permissions objects.

```

Table "content_perm"
  Column      |          Type          | Modifiers
-----+-----+-----
id           | bigint                 | not null
cp_type      | character varying(10)  | not null
username     | character varying(255) |
groupname    | character varying(255) |
cps_id       | bigint                 |
creator      | character varying(255) |
creationdate | timestamp without time zone |
lastmodifier | character varying(255) |
lastmoddate  | timestamp without time zone |
Indexes:
    "content_perm_pkey" PRIMARY KEY, btree (id)
    "cp_gn_idx" btree (groupname)
    "cp_os_idx" btree (cps_id)
    "cp_un_idx" btree (username)
Foreign-key constraints:
    "fkbd74b31676e33274" FOREIGN KEY (cps_id) REFERENCES content_perm_set(id)

```

`content_perm_set`: one-to-many mapping for content items and their permissions, with added metadata.

```

Table "content_perm_set"
  Column      |          Type          | Modifiers
-----+-----+-----
id           | bigint                 | not null
cont_perm_type | character varying(10)  | not null
content_id    | bigint                 |
creationdate  | timestamp without time zone |
lastmoddate   | timestamp without time zone |
Indexes:
    "content_perm_set_pkey" PRIMARY KEY, btree (id)
    "cps_content_idx" btree (content_id)
Foreign-key constraints:
    "fkbf45a7992caf22c1" FOREIGN KEY (content_id) REFERENCES content(contentid)

```

Clustering

`clustersafety`: normally, this table only contains one row. The value of the `safetynumber` is what Confluence uses to find out whether another instance is sharing its database without being part of the cluster.

```

Table "clustersafety"
  Column      | Type      | Modifiers
-----+-----+-----
clustersafetyid | bigint    | not null
safetynumber   | integer   |
Indexes:
    "clustersafety_pkey" PRIMARY KEY, btree (clustersafetyid)

```

System information

confversion used by the upgrade system to determine what to expect from the database, so as to negotiate upgrades.

```

Table "confversion"
  Column      | Type      | Modifiers
-----+-----+-----
confversionid | bigint    | not null
buildnumber   | integer   | not null
installdate   | timestamp without time zone |
versiontag    | character varying(255) |
creationdate  | timestamp without time zone |
lastmoddate   | timestamp without time zone |
Indexes:
    "confversion_pkey" PRIMARY KEY, btree (confversionid)
    "confversion_buildnumber_key" UNIQUE, btree (buildnumber)

```

plugindata: records which plugins have been installed, and when.
data is a blob of the actual plugin .jar file. This is principally cluster-related.

```

Table "plugindata"
  Column      | Type      | Modifiers
-----+-----+-----
plugindataid | bigint    | not null
pluginkey    | character varying(255) | not null
filename     | character varying(255) | not null
lastmoddate  | timestamp without time zone |
data         | bytea     |
Indexes:
    "plugindata_pkey" PRIMARY KEY, btree (plugindataid)
    "plugindata_filename_key" UNIQUE, btree (filename)
    "plugindata_pluginkey_key" UNIQUE, btree (pluginkey)

```

Spaces

spacegroups: this table is only used by the hosted environment.

```

Table "spacegroups"
  Column      | Type      | Modifiers
-----+-----+-----
spacegroupid  | bigint    | not null
spacegroupname | character varying(255) |
spacegroupkey | character varying(255) | not null
licensekey    | text      |
creator       | character varying(255) |
creationdate  | timestamp without time zone |
lastmodifier  | character varying(255) |
lastmoddate   | timestamp without time zone |
Indexes:
    "spacegroups_pkey" PRIMARY KEY, btree (spacegroupid)
    "spacegroups_spacegroupkey_key" UNIQUE, btree (spacegroupkey)

```

Table "spacepermissions"

Column	Type	Modifiers
permid	bigint	not null
spaceid	bigint	
permttype	character varying(255)	not null
permgrouppname	character varying(255)	
permusername	character varying(255)	
creator	character varying(255)	
creationdate	timestamp without time zone	
lastmodifier	character varying(255)	
lastmoddate	timestamp without time zone	

Indexes:

"spacepermissions_pkey" PRIMARY KEY, btree (permid)

"sp_permttype_idx" btree (permttype)

"sp_pgname_idx" btree (permgrouppname)

"sp_puname_idx" btree (permusername)

"sp_spaceid_idx" btree (spaceid)

Foreign-key constraints:

"fk9228242d1b7bfee" FOREIGN KEY (spaceid) REFERENCES spaces(spaceid)

spaces: information about the spaces themselves: key, human-friendly name and numeric ID.

Table "spaces"

Column	Type	Modifiers
spaceid	bigint	not null
spacename	character varying(255)	
spacekey	character varying(255)	not null
spacedescid	bigint	
homepage	bigint	
creator	character varying(255)	
creationdate	timestamp without time zone	
lastmodifier	character varying(255)	
lastmoddate	timestamp without time zone	
spacetype	character varying(255)	
spacegroupid	bigint	

Indexes:

"spaces_pkey" PRIMARY KEY, btree (spaceid)

"spaces_spacekey_key" UNIQUE, btree (spacekey)

"s_homepage_idx" btree (homepage)

"s_spacedescid_idx" btree (spacedescid)

"s_spacegroupid_idx" btree (spacegroupid)

Foreign-key constraints:

"fk9228242d1b7bfee" FOREIGN KEY (homepage) REFERENCES content(contentid)

"fk9228242d16994414" FOREIGN KEY (spacegroupid) REFERENCES spacegroups(spacegroupid)

"fk9228242d2c72d3d2" FOREIGN KEY (spacedescid) REFERENCES content(contentid)

Appearance

decorator: storage of custom display templates, for customising layouts.

Table "decorator"

Column	Type	Modifiers
decoratorid	bigint	not null
spacekey	character varying(255)	
decoratorname	character varying(255)	
body	text	
lastmoddate	timestamp without time zone	

Indexes:

"decorator_pkey" PRIMARY KEY, btree (decoratorid)

"dec_key_idx" btree (spacekey)

"dec_name_idx" btree (decoratorname)

Miscellaneous

`os_propertyentry`: for arbitrary association of entities and properties.

```
Table "os_propertyentry"
  Column |          Type          | Modifiers
-----+-----+-----
entity_name | character varying(125) | not null
entity_id   | bigint                 | not null
entity_key  | character varying(200) | not null
key_type    | integer                |
boolean_val | boolean                |
double_val  | double precision        |
string_val  | character varying(255) |
text_val    | text                   |
long_val    | bigint                 |
int_val     | integer                |
date_val    | timestamp without time zone |
Indexes:
    "os_propertyentry_pkey" PRIMARY KEY, btree (entity_name, entity_id, entity_key)
```

`bandana`: a catch-all persistence layer. It contains things like user settings and space- and global-level configuration data, and is used as storage by plugins such as the Dynamic Task List plugin. Essentially, for storing arbitrary data that doesn't fit anywhere else.

```
Table "bandana"
  Column |          Type          | Modifiers
-----+-----+-----
bandanaid | bigint                 | not null
bandanacontext | character varying(255) |
bandanakey | character varying(100) |
bandanavalue | text                   |
Indexes:
    "bandana_pkey" PRIMARY KEY, btree (bandanaid)
    "band_context_idx" btree (bandanacontext)
    "band_key_idx" btree (bandanakey)
```

`extrnltnks`: storage of referral links.

```
Table "extrnltnks"
  Column |          Type          | Modifiers
-----+-----+-----
linkid   | bigint                 | not null
contenttype | character varying(255) | not null
viewcount | integer                | not null
url       | character varying(255) | not null
contentid | bigint                 | not null
creator   | character varying(255) |
creationdate | timestamp without time zone |
lastmodifier | character varying(255) |
lastmoddate | timestamp without time zone |
Indexes:
    "extrnltnks_pkey" PRIMARY KEY, btree (linkid)
    "el_contentid_idx" btree (contentid)
Foreign-key constraints:
    "fk97c10fe78dd41734" FOREIGN KEY (contentid) REFERENCES content(contentid)
```

`hibernate_unique_key`: used by the high/low ID generator - the subsystem which generates our primary keys. Mess with this at the cost of being able to create objects.

```
Table "hibernate_unique_key"
  Column | Type | Modifiers
-----+-----+-----
next_hi | integer |
```

`indexqueueentries`: arbitrates full-content indexing across the system.

This table generally contains the last 12 hours or so of updates, to allow re-syncing of cluster nodes after restarts.

```
Table "indexqueueentries"
  Column      |          Type          | Modifiers
-----+-----+-----
entryid      | bigint                 | not null
creationdate  | timestamp without time zone |
type         | integer                |
handle       | character varying(255) |
Indexes:
  "indexqueueentries_pkey" PRIMARY KEY, btree (entryid)
```

keystore: used by the trusted apps framework to store the server's private key, and other servers' public keys.

```
Table "keystore"
  Column      |          Type          | Modifiers
-----+-----+-----
keyid        | bigint                 | not null
alias        | character varying(255) | not null
type         | character varying(32)  | not null
algorithm    | character varying(32)  | not null
keyspec      | text                   | not null
Indexes:
  "keystore_pkey" PRIMARY KEY, btree (keyid)
```

links: tracks links within the server (i.e. across and within spaces).

```
Table "links"
  Column      |          Type          | Modifiers
-----+-----+-----
linkid       | bigint                 | not null
destpagetitle | character varying(255) |
destspacekey | character varying(255) | not null
contentid    | bigint                 | not null
creator      | character varying(255) |
creationdate  | timestamp without time zone |
lastmodifier  | character varying(255) |
lastmoddate  | timestamp without time zone |
Indexes:
  "links_pkey" PRIMARY KEY, btree (linkid)
  "l_contentid_idx" btree (contentid)
  "l_destspacekey_idx" btree (destspacekey)
Foreign-key constraints:
  "fk45157998dd41734" FOREIGN KEY (contentid) REFERENCES content(contentid)
```

notifications: storage of page- and space-level watches.

```
Table "notifications"
  Column      |          Type          | Modifiers
-----+-----+-----
notificationid | bigint                 | not null
pageid        | bigint                 |
spaceid       | bigint                 |
username      | character varying(255) | not null
creator       | character varying(255) |
creationdate  | timestamp without time zone |
lastmodifier  | character varying(255) |
lastmoddate  | timestamp without time zone |
Indexes:
  "notifications_pkey" PRIMARY KEY, btree (notificationid)
  "n_pageid_idx" btree (pageid)
  "n_spaceid_idx" btree (spaceid)
Foreign-key constraints:
  "fk594acc88c38fbea" FOREIGN KEY (pageid) REFERENCES content(contentid)
  "fk594acc8b2dc6081" FOREIGN KEY (spaceid) REFERENCES spaces(spaceid)
```


pagetemplates: acts as the back-end of the templates feature.

```

Table "pagetemplates"
  Column      |      Type      | Modifiers
-----+-----+-----
templateid   | bigint         | not null
templatename | character varying(255) | not null
templatedesc | character varying(255) |
labels       | character varying(255) |
content      | text           |
spaceid      | bigint         |
prevver      | bigint         |
version      | integer        | not null
creator      | character varying(255) |
creationdate | timestamp without time zone |
lastmodifier | character varying(255) |
lastmoddate  | timestamp without time zone |
Indexes:
    "pagetemplates_pkey" PRIMARY KEY, btree (templateid)
    "pt_prevver_idx" btree (prevver)
    "pt_spaceid_idx" btree (spaceid)
Foreign-key constraints:
    "fkbc7ce96a17d4a070" FOREIGN KEY (prevver) REFERENCES pagetemplates(templateid)
    "fkbc7ce96ab2dc6081" FOREIGN KEY (spaceid) REFERENCES spaces(spaceid)

```

```

Table "trackbacklinks"
  Column      |      Type      | Modifiers
-----+-----+-----
linkid        | bigint         | not null
contenttype   | character varying(255) | not null
viewcount     | integer        | not null
url           | character varying(255) | not null
title         | character varying(255) |
blogname      | character varying(255) |
excerpt       | character varying(255) |
contentid     | bigint         | not null
creator       | character varying(255) |
creationdate  | timestamp without time zone |
lastmodifier  | character varying(255) |
lastmoddate   | timestamp without time zone |
Indexes:
    "trackbacklinks_pkey" PRIMARY KEY, btree (linkid)
    "tbl_contentid_idx" btree (contentid)
Foreign-key constraints:
    "fkf6977a478dd41734" FOREIGN KEY (contentid) REFERENCES content(contentid)

```

confancestors: used to speed up permissions checks, by allowing quick lookup of all a page's ancestors.

```

Table "confancestors"
  Column      | Type | Modifiers
-----+-----+-----
descendentid  | bigint | not null
ancestorid    | bigint | not null
ancestorposition | integer | not null
Indexes:
    "confancestors_pkey" PRIMARY KEY, btree (descendentid, ancestorposition)
Foreign-key constraints:
    "fk9494e23c37e35a2e" FOREIGN KEY (ancestorid) REFERENCES content(contentid)
    "fk9494e23cc45e94dc" FOREIGN KEY (descendentid) REFERENCES content(contentid)

```

Known Issues with Enterprise or Webhosting environments

When you attempt to run Confluence, you may get the following error:

```
java.security.AccessControlException: access denied (java.lang.RuntimePermission
accessDeclaredMembers)
    at
    java.security.AccessControlContext.checkPermission(AccessControlContext.java(Compiled Code))
        at java.security.AccessController.checkPermission(AccessController.java(Compiled Code))
        at java.lang.SecurityManager.checkPermission(SecurityManager.java(Compiled Code))
```

Some of the libraries Confluence relies on to function make use of features of the Java language that may be restricted by Java security policies. This does not normally cause any problems: the default security configuration of most application servers will happily run Confluence. However, in some shared-hosting or enterprise environments, security settings may be such that Confluence can not function.

The permissions required by Confluence to run are detailed in the sample policy file below. You may need to give this information to your systems administrator so that they can be deployed with the Confluence application.

```
grant codeBase "file:${catalina.home}/webapps/confluence/-" {
    permission java.security.AllPermission;
};

grant {
    permission java.lang.RuntimePermission "accessDeclaredMembers";
    permission java.lang.reflect.ReflectPermission "suppressAccessChecks";
    permission java.lang.RuntimePermission "defineCGLIBClassInJavaPackage";
};
```

Setting Up Public Access

Granting of permissions to use Confluence can be done on the basis of membership of a group, to a particular user, or to the 'Anonymous' user. There is not an actual user named 'Anonymous', it is just a name for a category of granted permissions.

In the security administration of Confluence, the 'Anonymous' user includes all logged-in users, *and* anonymous users who have not logged in. That is, if you allow the 'Anonymous' user to do something, you are allowing all users to do it.

To enable public access to a confluence space, you must give 'Anonymous' the following permissions:

- The global '**Use Confluence**' permission. This permission determines whether a user can access the Confluence installation at all, and is set by the site administrator in **Administration -> Global Permissions**.
- The relevant [space permissions](#). The 'Anonymous' user must have at least the 'View Space' permission for a space to be publicly accessible. You set these permissions via '**Browse Space**' -> '**Space Admin**' -> '**Permissions**'.

While these two permissions are the bare minimum necessary for public access to a space, you may wish to give 'Anonymous' additional permissions if you want a space to allow anonymous comments, or anonymous editing.

We **severely warn against** giving the 'Anonymous' user any administrative privileges, either within a space, or especially globally over the Confluence instance. Giving administrative privileges to untrusted users may lead to a serious security compromise of your site.

Related

Page: [Enabling or Disabling Public Signup](#)

Setting Up a Mail Session for the Confluence Distribution

Set up a mail session for the Confluence distribution to use Gmail as follows:

1. Stop Confluence.
2. Move (don't copy) activation-1.0.2.jar and mail-1.4.1.jar from <confluence-install>/confluence/WEB-INF/lib to <confluence-install>/lib.
3. Add the following to your server.xml file found in <confluence-install>/conf/ (add it just before the </Context> tag):

For Confluence 3.5.x

```
<Resource name="mail/GmailSMTPServer"
  auth="Container"
  type="javax.mail.Session"
  mail.smtp.host="smtp.gmail.com"
  mail.smtp.port="465"
  mail.smtp.auth="true"
  mail.smtp.user="yourEmailAddress@gmail.com"
  password="yourPassword"
  mail.smtp.starttls.enable="true"
  mail.transport.protocol="smtps"
  mail.smtp.socketFactory.class="javax.net.ssl.SSLSocketFactory"
/>
```

- Restart Confluence.
- Go to **Browse > Confluence Admin** and click on **Mail Servers**. Choose either **Edit an existing configuration**, or **Add a new SMTP mail server**.
- Edit the server settings as necessary, and set the **JNDI Location** as:

```
java:comp/env/mail/GmailSMTPServer
```

Note that the JNDI Location is case sensitive and must match the resource name specified in server.xml.

- Submit, and send a test email.

Troubleshooting SQL Exceptions

If you get an exception similar to those shown below, it is a good idea to increase the logging levels of your Confluence instance. If you request [Atlassian support](#), this additional logging will help us work out the cause of the error.

Increased logging levels will enable us to diagnose errors like these:

```
org.springframework.dao.DataIntegrityViolationException: (HibernateTemplate): data integrity
violated by SQL ''; nested exception is java.sql.BatchUpdateException: Duplicate entry '1234'
for key 1
at
org.springframework.jdbc.support.SQLStateSQLExceptionTranslator.translate(SQLStateSQLExceptionTrans
by: java.sql.BatchUpdateException: Duplicate entry '1234' for key 1
at com.mysql.jdbc.ServerPreparedStatement.executeBatch(ServerPreparedStatement.java:647)
```

or

```
(HibernateTemplate): data integrity violated by SQL ''; nested exception is
java.sql.BatchUpdateException: ORA-00001: unique constraint (CONFLUENCE.SYS_C0012345) violated
```

This document outlines the steps to take to increasing logging on your system.



Changing the logging levels via the Administration Console

With Confluence 2.7 and later, you can adjust logging levels at runtime via the Administration Console — read the [instructions](#). Below we tell you how to edit the log4j files directly.

- Open `confluence/WEB-INF/classes/log4j.properties` and uncomment the following lines. The double `##` lines are comments, leave them intact.

```
## log hibernate prepared statements/SQL queries (equivalent to setting  
'hibernate.show_sql' to 'true')  
#log4j.logger.net.sf.hibernate.SQL=DEBUG  
  
## log hibernate prepared statement parameter values  
#log4j.logger.net.sf.hibernate.type=DEBUG
```



If you can not locate these lines in your `log4j.properties` file, please add them to the end of it.

2. Restart Confluence.
3. Redo the steps that led to the error.
4. Zip up your logs directory and attach it your support ticket.
5. If you are using Oracle and received a **constraint error**, please ask your database administrator which *table* and *column* the constraint (that is, `CONFLUENCE.SYS_C0012345`) refers to and add that information to your support ticket.
6. Open `confluence/WEB-INF/classes/log4j.properties` again and remove the 4 lines you added in step 1. (The additional logging will impact performance and should be disabled once you have completed this procedure.)

RELATED TOPICS

[Enabling Detailed SQL Logging](#)

[Working with Confluence Logs](#)

[Troubleshooting failed XML site backups](#)