

# Installation

---

## Downloading Daisy

Packaged versions of Daisy can be found in the [distribution area](#)<sup>1</sup>. This includes everything required to run Daisy, except for:

- a Java Virtual Machine (JVM): version 1.4.2, version 5 (= 1.5) or higher required
- a MySQL database: version 4.0.20 or higher, or version 4.1.7 or higher required. Version 5.x is not yet supported (will be in Daisy 1.4)

If you don't have these already, the installation of these will be covered further on.

Consider subscribing to the [Daisy mailing list](#)<sup>2</sup> to ask questions and talk with fellow Daisy users and developers.

There is also information available about the [source code](#)<sup>3</sup>.

## Installation Overview

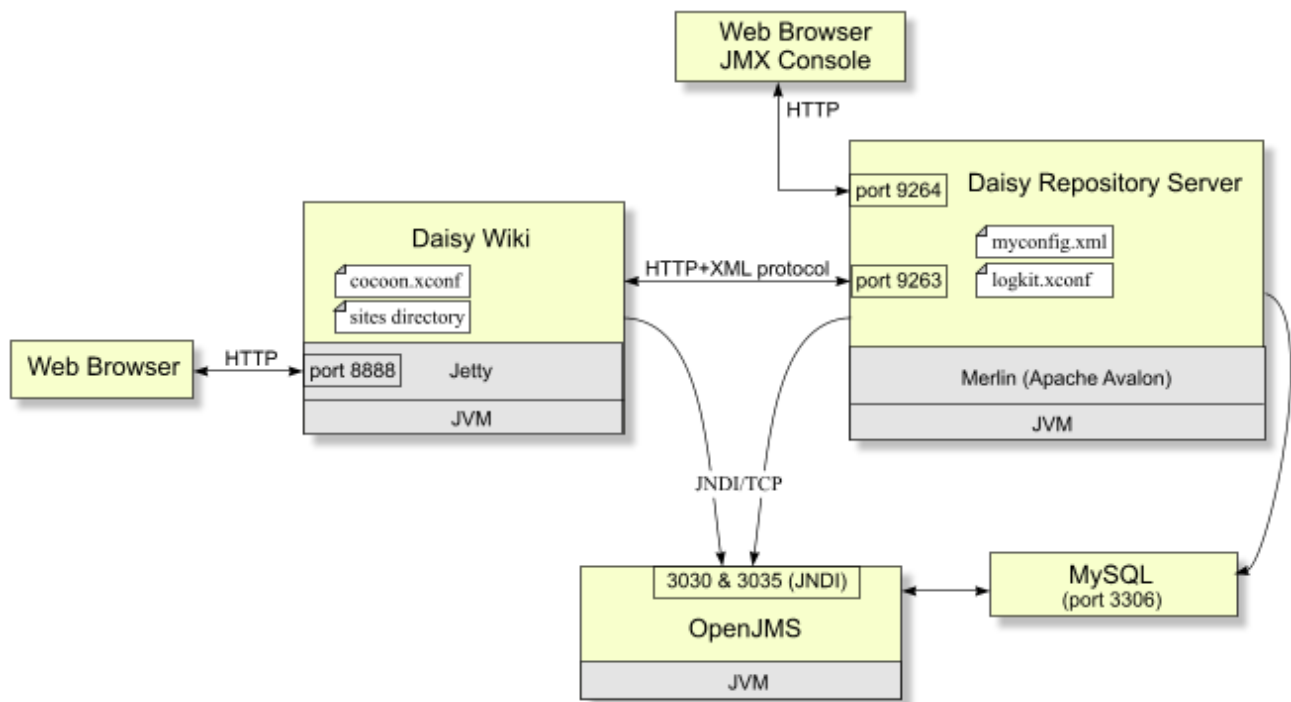
Daisy is a multi-tier application, consisting of a repository server and a publication layer. Next to those, a JMS server (OpenJMS) and a database server (MySQL) are required. All together, this means four processes, which can run on the same server or on different servers.

The Daisy binary distribution packs most of the needed software together, the only additional things you'll need is a Java Virtual Machine for your platform, and MySQL. All libraries and applications shipped with Daisy are the original, unmodified distributions that will be configured as part of the installation. We've only grouped them in one download for your convenience.

If you follow the instructions in this document, you can have Daisy up and running in less than an hour.

The diagram below gives an overview of the the setup. All shown port numbers are configurable of course.

- 
1. <http://svn.cocoondev.org/dist/daisy/>
  2. <http://lists.cocoondev.org/mailman/listinfo/daisy>
  3. [http://daisycms.org/daisydocs-1\\_3/152-cd.html](http://daisycms.org/daisydocs-1_3/152-cd.html) (Source Code)



## Platform Requirements

We have tested the Daisy installation on Windows 2000/XP, GNU/Linux and MacOSX. Other unixes like Solaris should also work, though we don't test that ourselves.

## Memory Requirements

By default, the Daisy Wiki and Daisy Repository Server are started with a maximum heap size of 128 MB each, OpenJMS uses the JVM default (64 MB). To this you need to add some overhead of the JVMs themselves, and then some memory for MySQL, the OS and its (filesystem) caches. This doesn't mean all this memory will be used, that will depend on usage intensity.

## Required knowledge

These installation instructions assume you're comfortable with installing software, editing configuration (XML) files, running applications from the command line, setting environment variables, and that sort of stuff.

## Can I use Oracle, PostgreSQL, MS-SQL, ... instead of MySQL? Websphere, Weblogic, Tomcat, ... instead of Jetty? What is this Merlin thing?

Daisy contains the necessary abstractions to support different database engines, though we currently only support MySQL. Users are welcome to contribute and maintain different databases (ask on the mailing list how to get started).

The Daisy Wiki webapp should be able to run in any servlet container (at least one that can run unpacked webapps, and as far as there aren't any Cocoon-specific issues), but we ship Jetty by default. For example, using Tomcat instead of Jetty is very simple and is described [on this page](#)<sup>4</sup>.

4. [http://daisycms.org/daisydocs-1\\_3/admin/208-cd.html](http://daisycms.org/daisydocs-1_3/admin/208-cd.html) (Deploying on Tomcat)

The Daisy Repository Server runs on top of a component runtime platform called [Merlin](#)<sup>5</sup>. Generally you won't be aware of this, but if you see "merlin" popping up in process names, that means it is basically the same as the Daisy Repository Server.

## Installing a Java Virtual Machine

Daisy requires either the Java SDK 1.4.2 or the Java JDK 5.0.

You can download the Java JDK 5.0 from [here on the Sun site](#)<sup>6</sup> (take the JDK, not the JRE). Install it now if you don't have it already.

After installation, make sure the JAVA\_HOME environment variable is defined and points to the correct location (ie, the directory where Java is installed). To verify this, open a command prompt or shell and enter:

```
For Windows:
%JAVA_HOME%/bin/java -version

For Linux:
$JAVA_HOME/bin/java -version
```

This should print out something like:

```
java version "1.4.2_xx"
```

or

```
java version "1.5.0"
```

## Installing JAI (Java Advanced Imaging) -- optional, Java 1.4 only

If you want images (especially PNG) to appear in PDFs, it is highly advisable to install JAI, which you can download from the [JAI homepage](#)<sup>7</sup>. Just take the "JDK Install" option, this will make JAI support globally available.

## Installing MySQL

Daisy requires one of the following MySQL versions:

- version 4.0.20 or higher (we're not making this number up: there have been various bug fixes or improvements between versions 4.0.0 and 4.0.20 on which Daisy depends)
- version 4.1.7 or a newer version from the 4.1.x series
- version 5.x is not yet supported (will be in Daisy 1.4)

The 4.1 version of MySQL has the big advantage that it supports UTF-8, so if you can choose, pick the 4.1.

MySQL can be downloaded from [mysql.com](#)<sup>8</sup>. Install it now, and start it (often done automatically by the install).

### NOTE

**Windows users** can take the "Windows Essentials" package. During installation and the configuration wizard, you can leave most things to their defaults. In particular, be sure to leave the "Database Usage" to "Multifunctional Database", and leave the TCP/IP Networking enabled (on port 3306). When it asks for the default character set, select "Best Support For

---

5. <http://avalon.apache.org>

6. <http://java.sun.com/j2se/1.5.0/download.jsp>

7. <http://java.sun.com/products/java-media/jai/>

8. <http://dev.mysql.com/downloads/>

Multilingualism" (this will use UTF-8). When it asks for Windows options, check the option "Include Bin Directory In Windows Path".

NOTE

**Linux users:** install the "MySQL server" and "MySQL client" packages. Installing the MySQL server RPM will automatically initialise and start the MySQL server. To enable UTF-8: after installation, shutdown MySQL (/etc/init.d/mysql stop), and edit or create the file /var/lib/mysql/my.cnf, and add the following:

```
[mysqld]
default-character-set=utf8
[mysql]
default-character-set=utf8
(note: don't do this if you already have data in your mysql database! See the MySQL
documentation)
Afterwards, start the server again by executing /etc/init.d/mysql start
```

## Creating MySQL databases and users

MySQL is used by both the Daisy Repository Server and OpenJMS. Therefore, we are now going to create two databases and two users.

Open a command prompt, and start the MySQL client as root user:

```
mysql -uroot -pYourRootPassword
```

On some systems, the root user has no password, in which case you can drop the -p parameter.

Now create the necessary databases, users and access rights by entering (or copy-paste) the commands below in the mysql client. What follows behind the IDENTIFIED BY is the password for the user, which you can change if you wish. The daisy@localhost entries are necessary because otherwise the default access rights for anonymous users @localhost will take precedence. If you'll run MySQL on the same machine as OpenJMS and the Daisy Repository Server, you only need the @localhost entries.

```
CREATE DATABASE daisyrepository;
GRANT ALL ON daisyrepository.* TO daisy@"%" IDENTIFIED BY "daisy";
GRANT ALL ON daisyrepository.* TO daisy@localhost IDENTIFIED BY "daisy";
CREATE DATABASE openjms;
GRANT ALL ON openjms.* TO openjms@"%" IDENTIFIED BY "openjms";
GRANT ALL ON openjms.* TO openjms@localhost IDENTIFIED BY "openjms";
```

## Extract the Daisy download

Extract the Daisy download. On Linux/Unix you can extract the .tar.gz file as follows:

```
tar xvzf daisy-<version>.tar.gz
```

NOTE

On non-Linux unixes, use the GNU tar version if you experience problems extracting.

On Windows, use the .zip download, which you can extract using a tool like WinZip.

Make sure that wherever you extract the download, none of the parent directories contains spaces in their names. For example, on Windows, do *not* extract Daisy inside your c:\Program Files directory.

After extraction, you will get a directory called `daisy-<version>`. This directory is what we will call from now on the **DAISY\_HOME** directory. You may set a global environment variable pointing to that location, or you can do it each time in the command prompt when needed.

## OpenJMS configuration

Open a command prompt or shell and set the following environment variables:

- `DAISY_HOME`, pointing to the directory where Daisy is installed
- `OPENJMS_HOME`, pointing to `<DAISY_HOME>/openjms`

Then go to the directory `<DAISY_HOME>/install`, and execute:

```
daisy-openjms-config
```

(Unix users might have to do `./daisy-openjms-config`, depending on their settings)

The application will ask a few database parameters and then update configuration files accordingly. If you have MySQL running on localhost and created the tables using default names and passwords as shown earlier, you can simply press enter on each question.

Then go to the directory `<OPENJMS_HOME>/bin`, and execute the following to create the OpenJMS database tables (the complete statement should be on one line):

```
For Linux:
dbtool.sh -create -config $OPENJMS_HOME/config/openjms.xml
          -Dlog4j.configuration=$DAISY_HOME/misc/openjms-log4j.properties

For Windows:
dbtool.bat -create -config %OPENJMS_HOME%\config\openjms.xml
          -Dlog4j.configuration=%DAISY_HOME%\misc\openjms-log4j.properties
```

This should print out as last line:

```
Successfully created tables
```

Now you can start OpenJMS: go to the directory `<OPENJMS_HOME>/bin`, and execute:

```
For Linux:
startup.sh

For Windows:
startup.bat
```

## Daisy Repository Server

### Initialiasing and configuring the Daisy Repository

Open a command prompt or shell and set an environment variable **DAISY\_HOME**, pointing to the directory where Daisy is installed.

Then go to the directory `<DAISY_HOME>/install`, and execute:

```
daisy-repository-init
```

Follow the instructions on screen. The installation will (1) initialias the database tables for the repository server and (2) create a Daisy data directory containing customized configuration files.

## Starting the Daisy Repository Server

Still in the same command prompt (or in a new one, but make sure `DAISY_HOME` is set), go to the directory `<DAISY_HOME>/repository-server/bin`, and execute:

```
daisy-repository-server <location-of-daisy-data-dir>
```

### WARNING

If you're on Windows and installed Java in a directory containing spaces in the name (for example, `c:\Program Files`), do the following: open the file `<DAISY_HOME>/repository-server/merlin/bin/merlin.bat` in a text editor. The third line counting from the end starts with `%JAVA_HOME%`, change this to `"%JAVA_HOME%"` (thus, put it between quotes).

In which you replace `<location-of-daisy-data-dir>` with the location of the daisy data directory created in the previous step.

The startup can take a few seconds, though nothing special is printed to the screen to indicate it is completely started (the prompt will not return)

### NOTE

OpenJMS and MySQL still need to be running, otherwise the repository server will be unable to start.

## Daisy Wiki

### Initialising the Daisy Wiki

Before you can run the Daisy Wiki, the repository needs to be initialised with some document types, a "guest" and "registrar" user, a default ACL configuration, etc.

Open a command prompt or shell and set an environment variable `DAISY_HOME`, pointing to the directory where Daisy is installed.

Go to the directory `<DAISY_HOME>/install`, and execute:

```
daisy-wiki-init
```

The program will start by asking a login and password, enter here the user created during the execution of `daisy-repository-init` (the default was `testuser/testuser`). It will also ask for the URL where the repository is listening, you can simply press enter here.

If everything goes according to plan, the program will now print out some informational messages and end with "Finished".

### Creating a Daisy Wiki Site

The Daisy Wiki has the concept of multiple sites, these are multiple views on top of the same repository. You need at least one site to do something useful with the Daisy Wiki, so we are now going to create one.

Open a command prompt or shell and set an environment variable `DAISY_HOME`, pointing to the directory where Daisy is installed.

Go to the directory `<DAISY_HOME>/install`, and execute:

```
daisy-wiki-add-site
```

The application starts by asking the same parameters as for daisy-wiki-init.

Then it will ask a name for the site. This should be a name without spaces. If you're inspirationless, enter something like "test" or "main".

Then it will ask for the sites directory location, for which the presented default should be OK, so just press enter.

## Starting the Daisy Wiki

Open a command prompt or shell and set an environment variable **DAISY\_HOME**, pointing to the directory where Daisy is installed.

Go to the directory <DAISY\_HOME>/daisywiki/bin, and execute:

```
daisy-wiki
```

This will start Jetty (a servlet container) with the webapp found in <DAISY\_HOME>/daisywiki/webapp.

NOTE

OpenJMS, MySQL, and the Daisy Repository Server should still be running of course.

## Finished!

Now you can point your web browser to:

```
http://yourhost:8888/daisy/
```

(Note the final slash on the end)

To be able to create or edit documents, you will have to change the login, you can use the user you created for yourself while running daisy-repository-init (the default was testuser/testuser).