

Mindquarry Installation Guide

Installing the Mindquarry Collaboration Server on Linux, Ubuntu

Lars Trieloff, Mindquarry GmbH

<lars.trieloff@mindquarry.com>

Alexander Klimetschek, Mindquarry GmbH

<alexander.klimetschek@mindquarry.com>

Revision History

05/02/2007	Lars Trieloff
Initial creation of the installation guide.	
19/02/2007	Lars Trieloff
Added troubleshooting section to the installation guide.	
23/02/2007	Lars Trieloff
More detailed information about installing Mindquarry on Suse Linux, with thanks to Hans-Georg Dück, instructions on allowing anonymous repository access.	
16/03/2007	Lars Trieloff
Operating system-specific installation instructions, reflects the new Mindquarry Installer,	
25/03/2007	Alexander Klimetschek
Updating for 1.1-beta release.	
13/06/2007	Alexander Klimetschek
Updating for 1.1 release.	
17/06/2007	Alexander Klimetschek
More information about installing and upgrading on windows.	
03/07/2007	Alexander Klimetschek
Updating for 1.1.1 release, adding general upgrade description and list of links and property changes for each version.	
05/07/2007	Lars Trieloff
Clarifications regarding installation of Apache 2 with XAMMP on Windows, thanks to forum user omminp. Instructions on using Mindquarry with SSL support thanks to forum user wolf.	
11/07/2007	Alexander Klimetschek
Updating for 1.1.2 release	
23/08/2007	Lars Trieloff
Instructions on setting up the mail server for receiving e-mails	

Quick Installation

1. Install Java
2. Apache 2, mod_perl, Subversion
3. Install Mindquarry
4. Start Mindquarry
5. Configure your mail server
6. Go to <http://localhost:8080/> and log in with the username "admin" and password "admin".
7. When this works, start Apache 2 and go to <http://localhost/> and use Mindquarry

More details for these steps are below.

Note

If you are already using Mindquarry, you should read the upgrade guide, to avoid losing your data. Only if you don't need your old data anymore, it's okay to delete your old installation and install from scratch.

Table of Contents

Installation Prerequisites	2
Installing Java 6	2
Installing Apache 2 with Subversion and mod_perl on Linux	2
Installing the Mindquarry Server on Linux	3
Installing Mindquarry with a binary package	3
Configuring Mindquarry	4
Advanced Configuration	5
Configuring Apache	5
Configuring your Mail Server	6
Using the embedded Mail Server	6
Configuring an external Postfix Mail Server	7
Starting and Stopping Mindquarry	8
A. Troubleshooting & FAQ	9
B. Upgrade Guide	10
Upgrading (without Migration)	10
List of releases with links and property changes	10
Upgrading from 1.0-M1 or 1.1-beta to 1.1	12

Installation Prerequisites

The Mindquarry Collaboration Server can be installed on any operating system, provided that the following prerequisites are met.

- Java 5 (or higher) Software Development Kit
- Apache 2
- Subversion as an Apache module
- mod_perl
- mod_rewrite (comes with Apache 2, but by default is not enabled on all Linux distributions)
- Access to an outgoing mail server, if you want advanced e-mail integration

Installing Java 6

If you do not already have Java 5 or higher installed, download the JDK 6 from Sun. It is available for Windows, Linux and Solaris.

For many Linux distributions there are packages of Java 5 or Java 6 available from the Linux distributors.

Procedure 1. Installing Java on Ubuntu Linux (Console Mode)

1. Open a terminal window
2. Type **sudo apt-get install sun-java5-jdk**. When asked for a password, enter your password

Installing Apache 2 with Subversion and mod_perl on Linux

Apache 2 is the HTTP server that is used to provide the Subversion repositories for file sharing. Additionally it can be used to secure the Mindquarry installation by adding SSL encryption. Apache 2 is available for download from the Apache Software Foundation. Many Linux distributions have pre-configured packages of Apache 2, Subversion and mod_perl available.

Make sure to install the `mod_dav_svn`, `mod_dav`, `mod_perl`, `mod_proxy`, `mod_proxy_http` and `mod_rewrite` modules into your Apache 2 web server. Usually this is done by having a following lines in the Apache 2 configuration file.

```
LoadModule dav_svn_module      modules/mod_dav_svn.so
LoadModule dav_module         modules/mod_dav.so
LoadModule proxy_module       modules/mod_proxy.so
LoadModule proxy_http_module   modules/mod_proxy_http.so
LoadModule perl_module        modules/mod_perl.so
LoadModule rewrite_module     modules/mod_rewrite.so
```

Procedure 2. Installing Apache 2, Subversion and mod_perl on Ubuntu

1. Open a terminal window
2. Type **sudo apt-get install apache2 libapache2-svn libapache2-mod-perl2**. When asked for a password, enter your password
3. Activate mod_perl by typing **sudo ln -s /etc/apache2/mods-available/perl.load /etc/apache2/mods-enabled/**
4. Activate Subversion by typing **sudo ln -s /etc/apache2/mods-available/dav.load /etc/apache2/mods-enabled/** and **sudo ln -s /etc/apache2/mods-available/dav_svn.load /etc/apache2/mods-enabled/**
5. Activate mod_proxy by typing **sudo ln -s /etc/apache2/mods-available/proxy.load /etc/apache2/mods-enabled/**
6. Activate mod_proxy_http by typing **sudo ln -s /etc/apache2/mods-available/proxy_http.load /etc/apache2/mods-enabled/**
7. Activate mod_rewrite by typing **sudo ln -s /etc/apache2/mods-available/rewrite.load /etc/apache2/mods-enabled/**
8. Test Apache's configuration by running **sudo apache2ctl configtest**.

Installing the Mindquarry Server on Linux

In order to install Mindquarry on Linux, the following steps are necessary:

Note

If you are upgrading, please follow the instructions in the upgrade guide, as your existing data could be overwritten if it is located inside the installation directory.

1. Download the Mindquarry Installer for your hardware architecture from the Mindquarry Download site
2. Open a terminal window
3. Type **sudo sh mindquarry-launcher-*.bin**. When asked for a password, enter your password

The installer will check for required software like Java, mod_perl and Subversion, ask you for some configuration parameters and install Mindquarry.

Mindquarry can then be started by running the `/opt/mindquarry/bin/mindquarry` script. Make sure to restart the Apache Web server to reflect the configuration changes made by Mindquarry. This is usually done using the `/etc/init.d/apache2 restart` command.

Installing Mindquarry with a binary package

In order to install Mindquarry manually with a binary package, the following steps are necessary:

1. Download a Mindquarry binary package for your operating system from the Mindquarry Download Repository
2. Unpack the binary package and copy it to a location of your choice. Mindquarry recommends `/opt/mindquarry` as the installation directory.
3. You have to follow the instructions in the configuration.

Mindquarry can then be started by running the `/opt/mindquarry/bin/mindquarry` script.

In the future there will be packages available for most Linux distributions that allow installation of Mindquarry using the operating system's package management.

Configuring Mindquarry

These configuration settings are for advanced use of Mindquarry. When you are using an installer, reasonable default already have been set, so you can skip this section safely.

All configuration of the Mindquarry Collaboration Server (but not the Apache environment and startup settings) is handled in `/opt/mindquarry/etc/mindquarry-webapplication.properties`. This file is a simple list of name-value-pairs. A line starting with a hash (#) is ignored, it contains a comment. All properties are documented. If you would like a more in-depth view of all the configuration, please read through the comments. For upgrading users you will find a list of new properties in each release below.

Most properties are tuned for running Mindquarry out of the box, you only have to configure three settings:

<code>mindquarry.title</code>	How your Mindquarry server is called. Use this to distinguish one Mindquarry installation from another. This could for instance be the name of your company or your workgroup.
<code>mindquarry.repos.uri</code>	The location where Mindquarry's file sharing repositories are located. Usually you need to replace <i>your.mindquarry.server</i> with the actual DNS name of the machine where Mindquarry is installed. With the default Apache and <code>mod_dav_svn</code> configuration, this is the server name + <code>/repos</code> . If your server is called <code>mindquarry.mycompany.com</code> , the value of this variable should be <code>http://mindquarry.mycompany.com/repos</code>

Note

Please note that the URL should not contain a port number (like `http://myserver:8080/repos`), because your Apache web server runs on port 80 and this is the default port for URLs starting with `http://`.

<code>mindquarry.server.url</code>	The location where Mindquarry is installed and available from the network via a browser, either through the <code>http://</code> or the <code>https://</code> protocol (depending on the configuration). If your server is called <code>mindquarry.mycompany.com</code> , the value of this variable should be <code>http://mindquarry.mycompany.com</code> . Accessing the server must be done via this exact URL - using <code>http://localhost</code> or the IP address instead will break things, because many absolute links inside the web interface are based on this configuration.
------------------------------------	---

Note

The URL is case-sensitive, so Mindquarry will warn you if you entered `http://`

MyMindQuarryServer.com in the configuration, but access it at `http://mymindquarryserver`.

The following settings can be left unchanged. They depict the data repositories, it is advisable to use a common directory with a large amount of available space eg. `/var/mindquarry`. An older default depicted to the `data` directory inside the installation, but this is *not* recommended since it is easier for upgrading to new versions when the data is separate from the server binaries. The three settings must have different directories.

<code>mindquarry.jcr.path</code>	Where Mindquarry's data files are stored, eg. <code>file:///var/mindquarry/repo</code> . This path must be absolute and start with <code>file://</code> .
<code>mindquarry.repos.path</code>	Where Mindquarry's file sharing repositories are stored, eg. <code>/var/mindquarry/docs</code> . This path can be relative.
<code>mindquarry.solr.path</code>	Where Mindquarry's search index is stored, eg. <code>/var/mindquarry/index</code> . This path can be relative.

5

Advanced Configuration

For advanced configuration parameters, the files `/opt/mindquarry/bin/mindquarry` and `/opt/mindquarry/etc/wrapper.conf` are important. The most important parameter for the first file is `RUN_AS_USER`. This parameter is commented out by default, but by removing the leading `#` and entering a user name, you can specify the user that should run Mindquarry.

For many Linux distributions it is advisable to set this parameter to the username of the user that runs the Apache web server, so that shared access to the created Subversion repositories is possible. The Mindquarry installer tries to read your Apache configuration and guess the username.

Example 1. Running Mindquarry as user apache

```
# If specified, the Wrapper will be run as the specified user.
# IMPORTANT - Make sure that the user has the required privileges to write
# the PID file and wrapper.log files. Failure to be able to write the log
# file will cause the Wrapper to exit without any way to write out an error
# message.
# NOTE - This will set the user which is used to run the Wrapper as well as
# the JVM and is not useful in situations where a privileged resource or
# port needs to be allocated prior to the user being changed.
RUN_AS_USER=apache
```

The file `/opt/mindquarry/etc/wrapper.conf` can be used to set memory limits for the Mindquarry installation. Usually the default settings does not to be changed, but installations having many users might need more memory.

Configuring Apache

If you are using an Installer, the Apache configuration already has been changed to handle Mindquarry, so you can safely skip this section unless you want to run Mindquarry on a virtual host or need more fine-grained configuration tuning.

Apache needs configuration for the following three purposes:

- Requests to the Mindquarry application should be handled by Mindquarry
- Requests to the Mindquarry file sharing repository should be handled by Subversion
- Requests to the Mindquarry file sharing repository should be authenticated by Mindquarry

Example 2. Running Mindquarry on a virtual host

```
NameVirtualHost *:80
DavLockDB /var/lock/apache
PerlRequire /opt/mindquarry/perl/Mindquarry/Authenticate.pm
<VirtualHost *:80>
  ServerName your.mindquarry.server
  <Location /repos>
    DAV svn
    SVNParentPath /var/mindquarry/data/docs
    SVNAutoversioning on

    PerlAuthenHandler Mindquarry::Authenticate
    PerlSetVar AuthBase "http://localhost:8080/files/authorise"
    AuthType Basic
    AuthName "Mindquarry Document Repository"
    Require valid-user
  </Location>
  ProxyRequests Off
  <Proxy *>
    Order Allow,Deny
    Allow from all
  </Proxy>
  RewriteEngine On
  RewriteCond %{REQUEST_URI} !^(repos|error)/(.*)$
  RewriteRule ^/(.*)$ http://localhost:8080/$1 [P,NC]
  ProxyPassReverse / http://localhost:8080/
</VirtualHost>
```

In this example only the `ServerName` directive needs to be changed, all other directives are perfect for a default installation. If you changed the path of the Mindquarry installation, you need to revise the `PerlRequire` directive. If you changed the `mindquarry.repos.path` configuration, you have to revise the `SVNParentPath` directive.

Restart the Apache Web Server to let these settings take effect.

Configuring your Mail Server

This step is not necessary for basic operation of Mindquarry, but if you rely on using advanced e-mail integration (sending files as attachment to the files section, starting and continuing conversations in the talk section via email), you should read this section. Mail server configuration differs from server to server, in this example we provide instructions for the Postfix mail server.

Mindquarry offers two options for configuring a mail server. If you do not have a mail server running on your Mindquarry server, the best choice is using the embedded mail server. If the server is already used to send and receive e-mails, this leads to a more difficult configuration. We cover this exemplarily for the postfix mail server.

Using the embedded Mail Server

Mindquarry includes an embedded mail server that receives e-mails, parses them and forwards them to the Mindquarry Collaboration Server. The embedded mail server has the same requirements as the Mindquarry Collaboration Server, namely Java 5 or Java 6. In order to get the embedded mail server up and running, following steps are necessary:

1. Configure the DNS to use your mail server as the mail exchange (MX) for the domain name your Mindquarry Server is running.
2. Set the password for the mail user by opening the `mindquarry-webapplication.properties` file and adding a parameter `mindquarry.mail.pwd`.
3. Download and unpack the `mindquarry-smtp` package from the Mindquarry Download Repository (during development, downloads are available at the Mindquarry Snapshot Repository)

The package to download is the tar.gz file for UNIX-based operating systems.

4. Go to the `bin` directory. Launch the `mindquarry-smtp` file under UNIX-based operating systems.

You need to provide four arguments to the program: the host name of the mail server, the address of the Mindquarry Collaboration Server, an username and a password. The synopsis for the command is:

```
mindquarry-smtp mailserver serverurl username password
```

Example 3. Launching mindquarry-smtp

```
# mindquarry-smtp mymindquarry.com http://mymindquarry.com mail m054pw
```

Note

The `mailserver` parameter should have the same value as the `mindquarry.mail.host` parameter in the `mindquarry-webapplication.properties` configuration file.

Configuring an external Postfix Mail Server

This is an advanced configuration option that can be used in scenarios where one mail server shall be used for receiving mail for ordinary users and as a mail proxy for the Mindquarry Collaboration Server. It is recommended to be already familiar with configuring Postfix before starting this procedure.

1. Configure the DNS to use your mail server as the mail exchange (MX) for the domain name your Mindquarry Server is running.
2. Set the password for the mail user by opening the `mindquarry-webapplication.properties` file and adding a parameter `mindquarry.mail.pwd`.
3. Download and uncompress the `mindquarry-talk-client` package from the Mindquarry Download Repository (during development, downloads are available at the Mindquarry Snapshot Repository)

The package to download is the JAR file (Java archive). It can be uncompressed using **unzip** or a similar program.

4. Add following lines at the end of the `/etc/postfix/main.cf` configuration file:

```
#configuration for mindquarry
virtual_mailbox_maps = hash:/etc/postfix/mindquarry

virtual_transport = mqmail

mqmail_destination_recipient_limit = 1
```

Caution

If the parameter `virtual_mailbox_maps` is already set in your configuration file, you should append `,hash:/etc/postfix/mindquarry` to the parameter's value instead of overwriting the parameter.

This tells postfix to use the virtual transport `mqmail` for all recipients listed in `/etc/postfix/mindquarry`. Additionally it tells postfix that this virtual transport is not able to handle more than one e-mail recipient at a time.

5. Create the file `/etc/postfix/mindquarry` with following content:

```
mymindquarry.com IGNORED
@mymindquarry.com IGNORED
```

Make sure to replace `mymindquarry.com` with the hostname of your mail server (this should be the same as the `mindquarry.mail.host` parameter in the `mindquarry-webapplication.properties` configuration file). Make also sure to include the line one with a leading `@` and once without.

Run the command **postalias** afterwards to create a hashed file that can be read by postfix faster:

```
# postalias /etc/postfix/mindquarry
```

6. Add following line at the end of the `/etc/postfix/master.cf` configuration file:

```
mqmail unix - n n - - pipe flags=FR user=nobody argv=/opt/mindquarry-mail/mqmail
```

This defines the virtual transport `mqmail`. This defines that the program `/opt/mindquarry-mail/mqmail` should be called by the user `nobody` for every mail that needs to be delivered to the Mindquarry Collaboration Server. You can adjust the username and the location of the `mqmail` script.

7. Install the Ruby interpreter. Ubuntu users can install Ruby by running **sudo apt-get install ruby**.
8. In the directory where you unpacked the `mindquarry-talk-client` package, for example `/opt/mindquarry-mail/`, create a script called `mqmail` with following content:

```
#!/bin/sh
ruby -I /opt/mindquarry-mail/ /opt/mindquarry-mail/post.rb" mymindquarry.com http://
```

This calls the Ruby script that uploads the mail message to the Mindquarry Collaboration Server. The parameters to the **ruby** program are following:

1. The directory of the unpacked `mindquarry-talk-client`
 2. The location of the `post.rb` file in the directory of the extracted `mindquarry-talk-client`
 3. The host name of the mail server receiving mails for Mindquarry. In the example, all mails that are not directed to an address ending with `@mymindquarry.com` will be rejected.
 4. The URL where the Mindquarry server can be reached. If the mail server and Mindquarry Collaboration Server are on one and the same machine, you can use `http://localhost:8080`
 5. The username of the `mail` user. This should not be changed.
 6. The password of the mail user. This is the same password as defined in the parameter `mindquarry.mail.pwd` in the `mindquarry-webapplication.properties` file.
9. Make the script executable.

```
# chmod +x mqmail
```

10. Restart postfix

```
# /etc/init.d/postfix restart
```

Starting and Stopping Mindquarry

Example 4. Starting Mindquarry under Linux/MacOSX

```
# /opt/mindquarry/bin/mindquarry start
```

Mindquarry can be stopped using the same script with the `stop` parameter. If you would like to follow the console output, start Mindquarry with the `console` parameter. This is useful for troubleshooting and might help tracing down problems.

Mindquarry can then be accessed using your web browser at `http://localhost/` (this depends on what was set as `mindquarry.server.url` and the domain name of the server, see the configuration section for more information). Log in as user `admin` with password `admin` to create new users and teams. If you do not see this page, you should stop the Mindquarry Collaboration Server, restart it with the console parameter and examine the error output.

A. Troubleshooting & FAQ

Support for Mindquarry is also available from the Mindquarry Support site and the Mindquarry Forum.

A.1.1. Is it possible to run Mindquarry on another port other than 8080?

Yes it is. Just add another parameter to `/opt/mindquarry/etc/wrapper.conf`, e.g.

```
wrapper.java.additional.4=-Djetty.port=8889
```

You need to change the Apache 2 configuration file accordingly.

A.1.2. I get file not found errors when restarting Mindquarry under Linux. There are messages about "Too many open files" in the `wrapper.log`. How to fix that?

This is most likely due to security limits set by your Linux distribution. The problem can be solved either by calling `ulimit -n 4096` (the value should be something larger than 4000) as root before starting Mindquarry or for persistent setting by modifying the `/etc/security/limit.conf` file by adding following lines:

<code>apache</code>	<code>soft</code>	<code>nofile</code>	<code>65536</code>
<code>apache</code>	<code>hard</code>	<code>nofile</code>	<code>65536</code>

This example assumes, Mindquarry is run as user `apache`. The Mindquarry installer will add these lines by default.

A.1.3. After some time of inactivity, Mindquarry complains about some files in the `/tmp` directory cannot be found.

Some operating systems as RedHat Enterprise use tools like `tmpwatch` for deleting files that are not used for a certain period of time. If this is the case then, deactivate this recurring task by deleting it from `/etc/cron.daily/tmpwatch`.

A.1.4. I cannot access the Mindquarry Document Repository using the Mindquarry Desktop Client. Looking at the Apache Error logs, I get "permission denied" errors.

Under some Linux configurations with SELinux, especially with RHEL, you have to extend the security context of the Apache Web Server by issuing following command (the path will be the one you have set as `mindquarry.repos.path`):

```
chcon -R -h -t httpd_sys_content_t /var/mindquarry/repos
```

A.1.5. How do I allow read-only anonymous access to the Subversion repository?

The Mindquarry Authentication perl module has a parameter called `AnonymousMethods` that specifies a list of HTTP methods that do not require authentication and enable anonymous access. Adding a line like the following allows anonymous read-only access to the repository.

```
PerlSetVar AnonymousMethods "GET PROPFIND OPTIONS REPORT"
```

A.1.6. I cannot synchronize with my desktop client. Accessing `http://server/repos/myteam/` and entering correct credentials also don't work. What's wrong?

If there is not yet a `DocumentRoot` set in your Apache config (look at Apache's global config files in `/etc/apache2/`) it introduces problems with the authentication, because the perl Authentication handler registered for `Location /repos` gets different URLs. Set `DocumentRoot` to some existing directory.

A.1.7.I want to use Mindquarry with SSL support, but do not have a server certificate signed by a trusted certificate authority. How do I set up Mindquarry?

First you need to configure Apache to use SSL encryption. There is plenty of documentation available at the Apache 2.0 documentation website. Secondly you need to make sure that the value of `mindquarry.server.url` starts with `https://`. If you are using a trusted certificate, everything is ok by now.

If you are using a self signed certificate, you have to add your certificate to the Java certification store of every client. You can use following command to import the certificate:

```
# $JAVA_HOME/bin/keytool -importcert -trustcacerts -file YOUR_CERT.cer -keystore li
```

The default password for the keystore is "changeit".

B. Upgrade Guide

Upgrading (without Migration)

Here are the short steps when doing a minor upgrade (eg. from 1.1 to 1.1.1) that does not involve a data or backend database migration. For upgrading special versions see the following section(s). There is no updating installer yet - if you re-run a new installer it will overwrite your existing settings and maybe your data, if it is contained within the installation directory.

To upgrade, you will basically have to update the `webapps` folder with the contents of the new `mindquarry-webapplication.war` and also update the file `etc/mindquarry-webapplication.properties` if new properties were added.

1. Go into your installation directory.
2. Make a backup of the existing `webapps` directory, eg. move it to `webapps-old`.
3. Download the `mindquarry-webapplication-VERSION.war` from <http://releases.mindquarry.org/com/mindquarry/webapp/mindquarry-webapplication/VERSION/> (see below for the list of versions and links).
4. Create a fresh new directory `webapps`.
5. Copy the downloaded file `mindquarry-webapplication-VERSION.war` into `webapps`, rename it to `mindquarry-webapplication-VERSION.zip`.
6. Extract `mindquarry-webapplication-VERSION.zip` in place.
7. Then remove the file `mindquarry-webapplication-VERSION.zip` from `webapps`.
8. Now download the new properties file `mindquarry-webapplication-VERSION.properties` from <http://releases.mindquarry.org/com/mindquarry/webapp/mindquarry-webapplication/VERSION/> (see below for the list of versions and links).
9. Add the new properties to your existing properties file -OR- copy your old settings into the new file and replace the existing one in `etc/mindquarry-webapplication.properties`. For a list of property changes in each release see below. This requires a bit of careful manual work - a simple Java tool called PropDiff can be helpful. Note that properties marked with **important** are needed for the server to startup and work properly, so make sure these have been set correctly.
10. Start the new server.
11. The log file `logs/wrapper.log` shows whether the start of the new version was successful.

List of releases with links and property changes

Release 1.1

Version	1.1
---------	-----

Webapplication Binary	mindquarry-webapplication-1.1.war
Webapplication Properties	mindquarry-webapplication-1.1.properties
Property changes:	(none, this list starts with 1.1)

Release 1.1.1

Version	1.1.1
Webapplication Binary	mindquarry-webapplication-1.1.1.war
Webapplication Properties	mindquarry-webapplication-1.1.1.properties
Property changes:	
mindquarry.jackrabbit.config	(new, important) Set to this value for now: <code>classpath:/com/mindquarry/jcr/jackrabbit/repository.xml</code>
mindquarry.bytes.encoding	(new, important) Regards a problem with user password encoding. Only to be set if you have data from previous installations (1.1 or lower). In those the system default encoding was used for encoding passwords in the JCR. If this changed (eg. having a different LANG environment variable or moving data from one system to the other), you could no longer login. In case you are upgrading from 1.1 put the "JVM OutputStream encoding" that is printed out by the Mindquarry server ($\geq 1.1.1$) on startup as value, eg. on Macs it typically would be: <code>mindquarry.bytes.encoding=MacRoman</code> , on Windows <code>mindquarry.bytes.encoding=Cp1252</code> . If you move data from one system to another, find out the default encoding of the old server and set it to the new server. If this value is not set, then UTF-8 is used as the default independent of the system default encoding.
mindquarry.mail.server.user	(new) Needed for password reset (and other mail features in the future). Set the user login for your mail server here (mail server address is configured via existing property <code>mindquarry.mail.server</code>)
mindquarry.mail.server.password	(new) Needed for password reset (and other mail features in the future). Set the user login password for your mail server here (mail server address is configured via existing property <code>mindquarry.mail.server</code>)

Release 1.1.2

Version	1.1.2
Webapplication Binary	mindquarry-webapplication-1.1.2.war
Webapplication Properties	mindquarry-webapplication-1.1.2.properties
Property changes:	
mindquarry.mail.fromAddress	(new) The From: address when sending password reset mails

Release 1.2-beta

Version	1.2-beta
Webapplication Binary	mindquarry-webapplication-1.2-beta.war
Webapplication Properties	mindquarry-webapplication-1.2-beta.properties
Property changes:	
<code>mindquarry.solr.login</code>	(removed) obsolete
<code>mindquarry.i18n.path</code>	(new) Path for translations. Useful for translators working on a translation or when users want to add another language file without re-installing. Can point to a directory with the translation files (<code>messages_LANG.xml</code> and <code>locales.xml</code>). Standard value is <code>resource://com/mindquarry/i18n</code>
<code>mail.mime.charset</code>	(new) must be set to UTF-8 (encoding for outgoing mails)
<code>mindquarry.mail.pwd</code>	(new) Password of the mail user used when uploading mails from the mail server to Mindquarry. For more information see the section called "Configuring your Mail Server".
<code>mindquarry.mail.host</code>	(new) hostname of the mail server that receives mails for Mindquarry. The default value is <code>{system-property:mindquarry.mail.host}</code> , this means identical to <code>mindquarry.mail.server</code> . For more information see the section called "Configuring your Mail Server".
<code>mindquarry.mail.fromAddress</code>	(modified) The From: address when sending password reset mails. You can now simply use <code>noreply@{system-property:mindquarry.mail.host}</code> which references the other property <code>mindquarry.mail.host</code> , that typically is identical. But you are free to set what your server requires here.
<code>mindquarry.logo</code>	(new, optional) Absolute path to a PNG file with max. 150px width and exactly 36px height that will be shown in the top-left. If in comments, the default will be the built-in Mindquarry logo

Upgrading from 1.0-M1 or 1.1-beta to 1.1

If you want to upgrade from our previous release (1.0-M1 or 1.1-beta) to 1.1, follow these steps. You won't need to download the full installer or install and configure Apache again.

1. You will need to use our migration tool if you don't want to lose your data. Note that the migration script requires the *old* running version of the server (1.0-M1 or 1.1-beta), so it needs to be applied *before* the update. How to use the migration tool:
 1. Stop the old server
 2. Make a backup of your data. Backup at least the directory that is specified under `mindquarry.jcr.path` in your `mindquarry-webapplication.properties`.
 3. Start the old server again. Make sure that nobody is using the server during the following migration process.

4. Unzip the migration program and call the following command:

```
sh migrate.sh
```

5. Stop the old server. Should the migration tool fail, please do not continue. Restore your backup and visit the Mindquarry forum for help.

2. Download `mindquarry-webapplication-1.1.war` (the server binaries without the installer)

3. Download the new version of the config file `mindquarry-webapplication-1.1.properties` and copy it to your `etc` directory.

4. Make a backup of your existing configuration file `mindquarry-webapplication.properties`, e.g. `mindquarry-webapplication-old.properties`.

5. Modify the new configuration file and add the settings from your old configuration file. Note that `mindquarry.jcr.path` now requires an absolute path starting with `file:///`. See the section about the configuration above.

6. Copy the modified new configuration file `mindquarry-webapplication-1.1.properties` to `mindquarry-webapplication.properties`.

7. Modify the configuration file `etc/wrapper.conf`: The important change is to modify the line

```
wrapper.java.additional.3=-Dmindquarry.config.dir=etc/mindquarry-webapplication.prop  
to
```

```
wrapper.java.additional.3=-Dmindquarry.config=etc/mindquarry-webapplication.properti
```

8. Remove the directory `webapps` in your installation (it contains the old application binaries), after making a backup.

9. Create a fresh new directory `webapps`.

10. Copy the downloaded file `mindquarry-webapplication-1.1.war` into `webapps`, rename it to `mindquarry-webapplication-1.1.zip`.

11. Extract `mindquarry-webapplication-1.1.zip` in place.

12. Then remove the file `mindquarry-webapplication-1.1.zip` from `webapps`.

13. Start the new server.

14. The log file `logs/wrapper.log` shows whether the start of the new version was successful.