**Overview**:

 This document will provide the complete procedure to install Redmine ([www.redmin.org](http://www.redmin.org)) bug tracking software onto a Tomcat web server using MS SQL Server as a database client.

**Procedure:**

1. Install Redmine
	1. Download the source code from: [download page](http://www.redmine.org/projects/redmine/wiki/Download).
2. Create an empty database and accompanying user
	1. Example for database “redmine” on MS SQL Server
	2. The database, login and user can be created within SQL Server Management Studio with a few clicks.
	3. If you prefer the command line option with SQLCMD, here’s a basic example:
		* 1. USE [master]
			2. GO
			3. -- Very basic DB creation
			4. CREATE DATABASE [REDMINE]
			5. GO
			6. -- Creation of a login with SQL Server login/password authentication and no password expiration policy
			7. CREATE LOGIN [REDMINE] WITH PASSWORD=N'redminepassword', DEFAULT\_DATABASE=[REDMINE], CHECK\_EXPIRATION=OFF, CHECK\_POLICY=OFF
			8. GO
			9. -- User creation using previously created login authentication
			10. USE [REDMINE]
			11. GO
			12. CREATE USER [REDMINE] FOR LOGIN [REDMINE]
			13. GO
			14. -- User permissions set via roles
			15. EXEC sp\_addrolemember N'db\_datareader', N'REDMINE'
			16. GO
			17. EXEC sp\_addrolemember N'db\_datawriter', N'REDMINE'
			18. GO
3. Database connection configuration
	1. Copy “config/database.yml.example” to “config/database.yml”.
	2. Modify the config file to match the production settings. Example for MS SQL Server below
		1. Example for a SQL Server database (default host localhost, default port 1433):
			1. production:
			2. adapter: sqlserver
			3. database: redmine
			4. username: redmine #should match the database user name
			5. password: redminepassword #should match the login password
4. Download, extract and install Ruby1.9.3 and DevelopmentKit
	1. <http://rubyinstaller.org/downloads/>
		1. Make sure to download the corresponding developmentkit
	2. Install DevelopmentKit (in the devkit root directory, do the following cmds)
		1. “ruby dk.rb init”
		2. “ruby dk.rb install”
5. Install Dependencies
	1. Install Bundler (<http://gembundler.com>)
		1. “gem install bundler”
		2. “bundle install --without development test”
	2. Install other dependencies
		1. RMagick
			* 1. “bundle install --without development test rmagick”
		2. Don’t forget to re-run “bundle install --without development test” after adding or removing adapters in the “config/database.yml” file
6. Download and extract required MySQL connector
	1. Download the following from online
		1. “mysql-connector-c-noinstall-6.0.2-win32.zip”
	2. Extract the mysql connector to:
		1. C:\**your-root-dir-choice\mysql-connector**
	3. Copy “libmysql.dll” from the directory of root\mysql-connector\lib
	4. Paste “libmysql.dll” into your bin directory of your Ruby install
		1. For me, it’s: ruby193\bin
7. Install mysql2 gem, and rebundle install other gems
	1. Install mysql2 gem
		1. “gem install mysql2 -- --with-mysql-dir=c:\**your-root\mysql-connector**”
	2. Bundle install rest of gems
		1. “bundle install --without development test rmagick”
8. Session store secret generation
	1. Redmine 1.4.x
		1. “rake generate\_session\_store”
	2. Redmine 2.x
		1. “rake generate\_secret\_token”
9. Database schema objects creation
	1. Create the database structure by running the following command under the application root directory:
		1. set RAILS\_ENV=production
		rake db:migrate
	2. It will create tables by running all migrations one by one then create the set of permissions and the application administrator account named “admin”
10. Database default data set
	1. Insert default configuration data in the database
		1. set RAILS\_ENV=production
		set REDMINE\_LANG=fr

rake redmine:load\_defualt\_data

1. File system permissions
	1. Windows users skip this sections
2. Test the installation
	1. Test using WEBrick web server
		1. Redmine 1.4.x
			* 1. “ruby script/server webrick -e production”
		2. Redmine 2.x
			* 1. “ruby script/rails server webrick -e production”
3. Logging into the application
	1. Use default administrator account
		1. Login: admin
		2. Password: admin
4. Configuration
	1. Redmine settings are defined in a file named “config/configuration.yml”

Redmine Installation with Tomcat

<http://www.redmine.org/projects/redmine/wiki/HowTo_install_Redmine_in_Apache_Tomcat>

1. Install necessary tools
	1. Install Jruby binary
		1. “aptitude install jruby1.2”
	2. Go into the redmine directory
		1. “cd redmine-VERSION”
	3. Test running Redmine outside of Tomcat Servlet
		1. “jruby ./script/server -e production”
	4. Install warbler gem
		1. “gem install warbler”
2. Building the .war file
	1. Run warbler in config mode
		1. “warble config”
		2. If not in that path
			1. “~/.gem/ruby/1.8/bin/warble config”
	2. Edit the wabler config
		1. “vim config/warble.rb”
	3. Change line in “config.dirs”
		1. “config.dirs = %w(app config lib log vendor tmp extra files lang)”
	4. Uncommend the line and add i18n and rack for packaging
		1. “config.gems += ["activerecord-jdbcmysql-adapter", "jruby-openssl", "i18n", "rack"]”
	5. Create a “session secret”
		1. “rake config/initializers/session\_store.rb”
	6. Run warble again
		1. “warble”
		2. “~/.gem/ruby/1.8/bin/warble”
		3. File called redmine-VERSION.war should be Redmine directory now
3. Place Redmine war file into your tomcat webapps directory
	1. “cp redmine-VERSION.war $TOMCAT\_HOME/webapps/redmine.war”
	2. Redmine should be running in a browser:
		1. <http://your-tomcat-host.name:8080/redmine>
		2. Will likely have to adjust port and host