

Installing Redmine on Oracle

- [Installing Redmine](#)
- [Requirements](#)
- [Installation](#)
- [SMTP server Configuration](#)
- [Backups](#)

Requirements

- Ruby on Rails:
 - Redmine 0.7.x requires Rails 2.0.2
 - Redmine devel [r1623](#) and above is compatible with Rails 2.1
- A database:
 - Oracle XE

Optional:

- SVN binaries (≥ 1.3), for repository browsing (must be available in your PATH)
- RMagick (Gantt export to a png image)

Note that Rails has some compatibility issues with ruby 1.8.7. The recommended ruby version is 1.8.6.

Installation of Ruby on Rails

There are two pieces of software you'll need in order to use Rails on Windows XP: the Ruby programming language with RubyGems installation package, Rails 1.0 (1.1 was recently released; this article addresses 1.0 only), and the Rails Oracle database library. To download and install Ruby with the Gems package installer, do the following.

1. Go to <http://rubyinstaller.rubyforge.org/>.
2. Download the "One-Click Ruby Installer".
3. Follow the installation instructions.

Once the Ruby programming platform is installed you can use RubyGems, a Ruby application that allows you to install Ruby frameworks very easily. To install the Rails application development framework, all you need to do is open a command-line prompt and type in a single command.

```
C:\> gem install rails --remote
```

Now that you have Ruby, RubyGems, and Rails installed you can install a specialized code library that will allow your Rails applications to talk to your Oracle database server. To download and install Oracle Rails library, go to

<http://rubyforge.org/projects/ruby-oci8>, download ruby-oci8-1.0.3-mswin32.rb to your C drive, and then using your command window, execute the following command.

```
C:\> ruby ruby-oci8-1.0.3-mswin32.rb
```

Assuming that everything went smoothly, you have now installed Ruby, Rails, and the Rails-Oracle connection library. You are now ready to create your first Ruby on Rails Web application.

The only thing you will need is the Oracle adapter, that's not included on Rails 2.0 . Download it at:

http://svn.rubyonrails.org/rails/adapters/oracle/lib/active_record/connection_adapters/

Copy the file oracle_adapter.rb to the folder: [/lib/ruby/gems/1.8/gems/activerecord-2.2.2/lib/active_record/connection_adapters](#)

Installation of Redmine

1. [Download](#) and extract the archive or [checkout](#) Redmine.
2. Create an user on Oracle named bduser, for example, with DBA privileges:

```
SQL> GRANT dba TO bduser IDENTIFIED BY bduser;
SQL> ALTER USER bduser DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp;
SQL> EXIT
```

3. Copy config/database.yml.example to config/database.yml and edit this file in order to configure your database settings for "production" environment.

```
production:
  adapter: oracle
  database: 127.0.0.1/1521:xe
  username: bduser
  password: bdpasswd
```

4. Change the tables of \ruby\apps\redmine\db\migrate\001_setup.rb to following:

```
create_table "users", :force => true do |t|
  t.column "login", :string, :limit => 30, :default => "", :null => true
  t.column "hashed_password", :string, :limit => 40, :default => "", :null => true
  t.column "firstname", :string, :limit => 30, :default => "", :null => true
  t.column "lastname", :string, :limit => 30, :default => "", :null => false
  t.column "mail", :string, :limit => 60, :default => "", :null => true
  t.column "mail_notification", :boolean, :default => true, :null => true
  t.column "admin", :boolean, :default => false, :null => true
  t.column "status", :integer, :default => 1, :null => false
  t.column "last_login_on", :datetime, :null => true
  t.column "language", :string, :limit => 2, :default => ""
  t.column "auth_source_id", :integer, :null => true
  t.column "created_on", :timestamp
  t.column "updated_on", :timestamp
end

create_table "projects", :force => true do |t|
  t.column "name", :string, :limit => 30, :default => "", :null => false
  t.column "description", :text, :default => nil, :null => true
  t.column "homepage", :string, :limit => 60, :default => ""
  t.column "is_public", :boolean, :default => true, :null => false
  t.column "parent_id", :integer
  t.column "projects_count", :integer, :default => 0
```

```
t.column "created_on", :timestamp
t.column "updated_on", :timestamp
end
```

5. With these changes, you can delete the following file:
ruby\apps\redmine\db\migrate\087_change_projects_description_to_text.rb

6. Alter the file above with the structure:

ruby\apps\redmine\db\migrate\091_change_changesets_revision_to_string.rb

```
class ChangeChangesetsRevisionToString < ActiveRecord::Migration
  def self.up
    change_column :changesets, :revision, :string
  end

  def self.down
    change_column :changesets, :revision, :integer
  end
end
```

7. The problem with UTF-8 on Oracle is can be resolved by adding a line on the beginning of config/environment.rb:

```
ENV['NLS_LANG']='american_america.AL32UTF8'
```

8. Create the database structure, by running the following command under the application root directory:

```
rake db:migrate RAILS_ENV="production"
```

It will create tables and an administrator account.

9. Insert configuration data in database, manually by specifying roles, permissions, types of tickets and so on. DON'T LOAD DEFAULT DATA.

```
rake redmine:load_default_data RAILS_ENV="production"
don't use this!!! It will fail!
```

10. Test the installation by running WEBrick web server:

```
ruby script/server -e production
```

Once WEBrick has started, point your browser to <http://localhost:3000/>. You should now see the application welcome page.

10. Use default administrator account to log in:

- login: admin
- password: admin

You can go to Admin & Settings to modify application settings.

SMTP server Configuration

In `config/environment.rb`, you can set parameters for your SMTP server:

- `config.action_mailer.smtp_settings`: SMTP server configuration
- `config.action_mailer.perform_deliveries`: set to `false` to disable mail delivering

Don't forget to restart the application after any change.

Backups

Redmine backups should include:

- data (stored in your redmine database)
- attachments (stored in the `files` directory of your Redmine install)

Here is a simple shell script that can be used for daily backups (assuming you're using a mysql database):

```
# Database
/usr/bin/mysqldump -u <username> -p <password> <redmine_database> |
gzip > /path/to/backup/db/redmine_`date +%y_%m_%d`.gz

# Attachments
rsync -a /path/to/redmine/files /path/to/backup/files
```

Special Thanks to **GUILHERME SCHNEIDER**, who became with light where there were only shadows in this thread: <http://www.redmine.org/boards/1/topics/show/1747#message-1763>

NOTE: This works fine with my enviroment, but it needs more test to validate the structure of issues tracking.

REFERENCES:

<http://www.redmine.org/wiki/redmine/RedmineInstall>

<http://web-loot.blogspot.com/2008/05/rails-20-e-oracle-xe.html>

<http://blog.andrewbeacock.com/2007/06/how-to-change-your-rails-database.html>

<http://michaelmarth.blogspot.com/2006/07/ruby-on-rails-unicode-and-oracle.html>

<http://www.oracle.com/technology/pub/articles/mearelli-optimizing-oracle-rails.html>

<http://www.oracle.com/technology/pub/articles/haefel-oracle-ruby.html>