



Community Experience Distilled

# Mastering Redmine

A comprehensive guide with tips, tricks and best practices, and an easy-to-learn structure

Andriy Lesyuk

[PACKT] open source\*  
PUBLISHING community experience distilled

# Mastering Redmine

A comprehensive guide with tips, tricks and best practices, and an easy-to-learn structure

**Andriy Lesyuk**

**[PACKT]** open source   
PUBLISHING community experience distilled  
BIRMINGHAM - MUMBAI

# Mastering Redmine

Copyright © 2013 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: January 2013

Production Reference: 1160113

Published by Packt Publishing Ltd.  
Livery Place  
35 Livery Street  
Birmingham B3 2PB, UK.

ISBN 978-1-84951-914-4

[www.packtpub.com](http://www.packtpub.com)

Cover Image by J. Blaminsky ([milak6@wp.pl](mailto:milak6@wp.pl))

# Credits

**Author**

Andriy Lesyuk

**Project Coordinator**

Abhishek Kori

**Reviewers**

Ludovic Gasc

Lester Martin

Mykhaylo Sorochan

Giovanni Toraldo

**Proofreaders**

Mario Cecere

Claire Cresswell-Lane

**Acquisition Editors**

Pramila Balan

Wilson D'souza

**Indexers**

Rekha Nair

Hemangini Bari

**Lead Technical Editor**

Arun Nadar

**Graphics**

Aditi Gajjar

Sheetal Aute

Valentina D'souza

**Technical Editors**

Sharvari Baet

Kaustubh S. Mayekar

Kirti Pujari

Devdutt Kulkarni

**Production Coordinators**

Shantanu Zagade

Conidon Miranda

**Cover Work**

Shantanu Zagade

# About the Author

**Andriy Lesyuk** is an open source evangelist, an enthusiastic, and passionate developer with more than 14 years of experience. He is skilled in Ruby, PHP, Perl, C, and more. His primary areas of interest are web development and Linux system development. He is also the author of more than 20 open source plugins for Redmine. He lives and works in Ivano-Frankivsk, Ukraine. His website is [www.andriylesyuk.com](http://www.andriylesyuk.com).

Andriy started his career as an engineer in Ivano-Frankivsk National Technical University of Oil and Gas, where he grew to be the Head of the Software and Networking Laboratory. For some time he worked as a freelancer developing custom Redmine plugins for companies world-wide, the most famous of which is oDesk. Recently, he joined the Kayako team which develops the world's leading helpdesk solution.

---

First, I would like to thank my wife, Lena Lesyuk, for support and patience. Special thanks to the Packt Publishing team, who assisted me in creating this book, especially to Sayama Wagh, Wilson D'souza, Pramila Balan, Abhishek Kori, Arun Nadar, Sharvari Baet, Devdutt Kulkarni, Kirti Pujari, Kaustubh S. Mayekar, and many others (Sorry, guys, I don't know all your names). And very much thanks to Redmine developers for their great work, especially to Jean-Philippe Lang, Eric Davis, and Toshi Maruyama.

---

# About the Reviewers

**Ludovic Gasc** is a senior software integration engineer at Eyepea, a highly renowned Open Source VoIP and Unified Communications company in Europe.

Over five years, Ludovic Gasc has managed more than a hundred projects using Redmine in a number of different fields (telephony, webapp development, education) and for a wide variety of purposes (support, coaching, development and so on).

He is also the author of one Redmine plugin and has published several patches.

**Lester Martin's** 20 year career showcases his accomplishments as a software development manager, architect, and programmer. His skills and experience are centered on Java, .NET, and e-commerce/web technologies. A self-described player/coach, he enjoys helping others grow in their software development careers while continuing to learn himself. He holds Sun Certified Java Programmer and Enterprise Architect credentials.

**Mykhaylo Sorochan**, PhD, has wide experience in Information Technologies ranging from software development to project management. He started from working on C++ console applications for Unix and now develops web applications in Ruby with Ruby on Rails framework. Besides professional interests in computer aided software engineering, domain specific languages, metaprogramming, and software transformation, his personal interests include such as science fiction, psychology, cycling, reading, and continuous acquiring of new knowledge. Currently holds a project manager position at the Sphere Consulting, Inc. - [www.sphereinc.com](http://www.sphereinc.com). In his free time, he supports Russian-speaking Redmine community - [www.redprojects.net](http://www.redprojects.net).

**Giovanni Toraldo** started to mess around with Linux and free software during his early years at school, developing hobbyist websites with free CMS and maintaining the official Italian support site of PHP-Fusion. After a few unsatisfactory years at university, he decided to start working as a System Administrator and Web Developer. Nowadays, he has developed skills in Linux systems administration, and has taken part in development teams actively using Redmine for years. He is also the author of the *OpenNebula* book released in mid-2012.

# www.PacktPub.com

## Support files, eBooks, discount offers and more

You might want to visit [www.PacktPub.com](http://www.PacktPub.com) for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

## Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

## Free Access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.



*I dedicate this book to my lovely wife, Lena Lesyuk.  
Her support and patience made it possible.*



# Table of Contents

<b>Preface</b>	<b>1</b>
<hr/>	
<b>Chapter 1: Getting Familiar with Redmine</b>	<b>5</b>
<hr/>	
<b>What is Redmine?</b>	<b>6</b>
Why does Redmine succeed	7
<b>Walk through the Redmine interface</b>	<b>8</b>
<b>MySQL, PostgreSQL, or SQLite</b>	<b>14</b>
<b>Textile or Markdown?</b>	<b>15</b>
<b>Selecting a Source Control Management (SCM)</b>	<b>16</b>
<b>Selecting a web server and an application server</b>	<b>18</b>
<b>Redmine or ChiliProject?</b>	<b>21</b>
<b>Recent versions of Redmine and ChiliProject</b>	<b>24</b>
Redmine versions	24
ChiliProject versions	25
<b>Helping Redmine</b>	<b>25</b>
<b>Summary</b>	<b>27</b>
<hr/>	
<b>Chapter 2: Installing Redmine</b>	<b>29</b>
<hr/>	
<b>Preface</b>	<b>30</b>
<b>Installing Redmine from the package</b>	<b>33</b>
Installing Redmine and MySQL server	33
Configuring MySQL server package	34
Configuring Redmine package	35
Installing Apache and Passenger	37
Connecting Redmine and Apache	38
Verifying and completing installation	40
Conclusion	41

<b>Installing Redmine from sources</b>	<b>42</b>
Downloading and installing Redmine	42
Installing RubyGems and Bundler	43
Resolving Bundler errors	43
Installing MySQL client and ImageMagick	44
Installing dependencies using Bundler	44
Installing MySQL server, Apache, and Passenger	45
Setting up the database	45
Finalizing Redmine installation	46
Configuring Apache	46
<b>Using TurnKey Redmine appliance</b>	<b>48</b>
<b>Using BitNami Redmine stack</b>	<b>52</b>
<b>Maintaining Redmine</b>	<b>60</b>
<b>Summary</b>	<b>62</b>
<b>Chapter 3: Configuring Redmine</b>	<b>63</b>
<b>The first thing to fix</b>	<b>63</b>
<b>Administration settings</b>	<b>64</b>
General	65
Cache formatted text	65
Other tips	66
Display	66
Default language	67
Using Gravatar user icons	67
Default Gravatar image	67
Display attachment thumbnails	68
Authentication	68
Allow OpenID login and registration	69
LDAP authentication	70
Enable REST web service	71
E-mail integration	72
Sending e-mails	72
Receiving e-mails	78
Repositories integration	85
Configuring repositories	85
<b>Troubleshooting</b>	<b>97</b>
<b>Summary</b>	<b>102</b>
<b>Chapter 4: Issue Tracking</b>	<b>103</b>
<b>Creating an issue</b>	<b>104</b>
Issue categories	108
Issues and project versions	110
<b>Issue page</b>	<b>110</b>
Subtasks	112

Related issues	113
<b>Issue list</b>	<b>115</b>
Custom queries	118
<b>Updating issue</b>	<b>121</b>
Bulk editing	122
<b>Getting updates on issues</b>	<b>124</b>
<b>Issue related settings</b>	<b>125</b>
Enabling or disabling issue tracking	125
Configuring issue tracking globally	126
Issues and repository integration	128
<b>Summary</b>	<b>130</b>
<b>Chapter 5: Managing Projects</b>	<b>131</b>
<b>Modules</b>	<b>132</b>
Issue tracking	132
Roadmap	132
Version page	135
News	136
Documents	138
Files	140
Wiki	141
Repository	145
Forums	151
Calendar	155
Gantt	156
<b>Global configuration</b>	<b>158</b>
<b>Creating a project</b>	<b>160</b>
<b>Basic project tabs</b>	<b>163</b>
Overview	163
Activity	165
<b>Configuring projects</b>	<b>167</b>
Information	168
Modules	168
Members	169
Versions	170
Wiki	173
<b>Closing a project</b>	<b>174</b>
<b>Project maintenance best practices</b>	<b>175</b>
Custom queries	175
<b>Project list</b>	<b>176</b>

<b>Adminstrating projects</b>	<b>177</b>
Copying projects	178
Archiving	179
<b>Summary</b>	<b>180</b>
<b>Chapter 6: Time Tracking</b>	<b>181</b>
<b>Using Redmine time tracking</b>	<b>182</b>
<b>Defining activity</b>	<b>182</b>
<b>Tracking time</b>	<b>184</b>
Tracking time using commit messages	188
<b>Checking spent time</b>	<b>189</b>
Time spent for issues	189
Time spent for versions	190
Time spent for projects	190
Project activity	191
Your time entries	191
<b>Generating reports</b>	<b>191</b>
Details tab	193
Report tab	194
Summary	195
<b>Chapter 7: Text Formatting</b>	<b>197</b>
<b>Formatting text in Redmine</b>	<b>198</b>
Wiki toolbar	200
Preview	201
Placing the linked images	201
<b>Wiki formatting syntax</b>	<b>202</b>
Basics	202
Quotes	206
Lists	207
Images	208
Links	209
Normal links	209
Internal links	210
Code	216
Tables	218
Macros	220
Table of contents	220
Collapsed block	221
Thumbnail	221
Include	222
Child pages	222
Hello world	222
Macro list	222

---

Advanced syntax	223
Alignment	223
Padding	224
Custom style and language	224
Textile span	225
Disabling element	225
Table advanced syntax	225
<b>Quick syntax reference</b>	<b>226</b>
<b>Summary</b>	<b>228</b>
<b>Chapter 8: Access Control and Workflow</b>	<b>229</b>
<b>Roles</b>	<b>230</b>
Permissions	232
Project block	233
Forums block	233
Calendar block	234
Documents block	234
Files block	235
Gantt block	235
Issue tracking block	235
News block	238
Repository block	238
Time tracking block	239
Wiki block	240
<b>Permissions report</b>	<b>241</b>
<b>Trackers</b>	<b>242</b>
<b>Issue statuses</b>	<b>245</b>
<b>Workflow</b>	<b>246</b>
Issue status transitions	246
Issue fields permissions	248
Quick functions	249
Copying configuration	250
Summary link	250
<b>Modifying workflow</b>	<b>251</b>
Adding roles	251
Adding trackers	252
Adding issue statuses	253
<b>Practical example</b>	<b>254</b>
<b>Summary</b>	<b>256</b>
<b>Chapter 9: Personalization</b>	<b>257</b>
<b>Automatic account creation</b>	<b>258</b>
OpenID	258
LDAP	261
<b>Gravatar</b>	<b>261</b>
<b>Personal page</b>	<b>265</b>

---

Issues assigned to me	267
Watched issues	267
Spent time	268
Documents	268
Reported issues	269
Latest news	270
Calendar	270
<b>Getting updates</b>	<b>271</b>
Notification settings	271
Watching	273
News feeds	274
<b>Personalizing issue lists</b>	<b>274</b>
<b>Summary</b>	<b>275</b>
<b>Chapter 10: Plugins and Themes</b>	<b>277</b>
<b>Looking up plugins</b>	<b>278</b>
Official registry	278
Github	281
<b>Installing a plugin</b>	<b>283</b>
Uninstalling a plugin	284
<b>Plugins review</b>	<b>285</b>
Meta	286
Exception Handler	288
Mylyn Connector	289
Screenshot Paste	293
Lightbox	294
Code Review	296
Niko Cale	299
Stuff To Do	302
<b>Themes</b>	<b>304</b>
Installing themes	304
Themes review	305
Basecamp	306
Modula Mojito	307
A1	308
Highrise	309
<b>Summary</b>	<b>310</b>

---

<b>Chapter 11: Customizing Redmine</b>	<b>311</b>
<b>Custom fields</b>	<b>312</b>
Custom field options	314
Required	315
Visible	315
Editable	315
For all projects	315
Searchable	315
Used as a filter	316
Custom field formats	317
Text	317
Long text	317
Integer	317
Float	317
List	318
Date	319
Boolean	319
User	319
Version	320
Customized object types	320
Issues	320
Spent time	321
Projects	321
Versions	321
Users	322
Groups	322
Activities (time tracking)	323
Issue priorities	323
Document categories	323
<b>Customizing with Textile</b>	<b>323</b>
Warnings and other boxes	324
Icons for text	324
Table-based layout	325
<b>Customizing themes</b>	<b>325</b>
<b>Customizing with plugins</b>	<b>326</b>
Writing a simple plugin	327
Customizing view files	328
Using hooks	329
<b>Summary</b>	<b>331</b>
<b>Index</b>	<b>333</b>

---



# Preface

This book describes the functionality and capabilities of Redmine, reveals its secrets and gives tips on how to use it. The book contains all the information needed to install, configure, use, and master this application. Redmine is a very powerful and an extremely flexible project management tool and an issue tracker. It's free, open source, built on the popular Ruby on Rails framework and has a strong community.

## What this book covers

*Chapter 1, Getting Familiar with Redmine*, prepares us for the next chapters by briefly going through the Redmine interface concept and reviewing pluggable components of the application.

*Chapter 2, Installing Redmine*, guides on how to install Redmine in four different ways, each of which is suitable for different purposes.

*Chapter 3, Configuring Redmine*, reviews the configuration options, which are available in the Settings section of the Administration area and covers the advanced options obscured behind them.

*Chapter 4, Issue Tracking*, reviews what makes Redmine one of the best issue trackers, also giving heed to the configuration options related to issue tracking.

*Chapter 5, Managing Projects*, covers the major part of Redmine functionality, which is related to projects, and shows, why this is one of the best applications for project hosting.

*Chapter 6, Time Tracking*, describes time tracking capabilities of Redmine and guides how to generate time reports.

*Chapter 7, Text Formatting*, is a comprehensive tutorial for the Redmine rich text formatting syntax, which is based on Textile and gives formatting powers to Redmine's Wiki.

*Chapter 8, Access Control and Workflow*, is aimed at enlightening the permissions system and the issue life cycle by explaining what is the role, the tracker, the issue status, and how are they connected.

*Chapter 9, Personalization*, gives hints on how to make Redmine more comfortable for users, and helps users to ensure that important and interesting information will be delivered to them.

*Chapter 10, Plugins and Themes*, guides on how to find plugins for a particular Redmine version, covers installation of plugins and themes and reviews some interesting plugins.

*Chapter 11, Customizing Redmine*, shows the power of custom fields and guides how to customize Redmine without breaking upgrade compatibility.

## What you need for this book

For this book, you need to have access to a Redmine installation (or you will need to install it). It's better (but not required), if you are an administrator of the installation.

The book describes Redmine 2.2.x. However, versions from 1.4.x to 2.1.x should be fine (they just miss some minor functionality).

## Who this book is for

This book is for anyone, who already uses or plans to use Redmine. But readers should consider, that most reviewed topics are specific to the software industry (while Redmine can be used for other industries as well). As the book also reviews topics, which require privileged access, it will be especially useful for project managers and Redmine administrators. No prior knowledge of Redmine is required.

## Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "We can include other contexts through the use of the `include` directive."



A block of code is set as follows:



```
<VirtualHost *:80>
  RailsEnv production
  PassengerAppRoot /opt/redmine/redmine-2.2.0
  DocumentRoot /opt/redmine/redmine-2.2.0/public
  <Directory "/opt/redmine/redmine-2.2.0/public">
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
```

Any command-line input or output is written as follows:

```
$ RAILS_ENV=production rake db:migrate
$ RAILS_ENV=production rake redmine:load_default_data
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "clicking the **Next** button moves you to the next screen".

 Warnings or important notes appear in a box like this. 

 Tips and tricks appear like this. 

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book – what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in the text or the code – we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

## Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

## Questions

You can contact us at [questions@packtpub.com](mailto:questions@packtpub.com) if you are having a problem with any aspect of the book, and we will do our best to address it.

# 1

## Getting Familiar with Redmine

When we are about to try a new web application we ask experts, who are familiar with it, what would they recommend us to use, for example, which database backend, which platform, and many more. That's what this chapter will try to do, it will help you learn what options are available and which one fits your needs better.

In this chapter you will also find a very short description of the **Redmine** interface. It's described in short because I believe it is easy to learn. You will be able to play with it more in the next chapters where we'll discuss how to install and use Redmine. So why this chapter should be of interest to you? The power of Redmine is in its components; some components influence performance, others influence functionality. I'm quite sure that even experienced users will discover new options in this chapter and may decide to switch to or utilize them.

In this chapter, we will cover the following topics:

- What is Redmine?
- Walking through the Redmine interface
- MySQL, PostgreSQL, or SQLite
- Textile or Markdown
- Selecting a **Source Control Management (SCM)**
- Selecting a web server and an application server
- Redmine or ChiliProject?
- Recent versions of Redmine and ChiliProject
- Helping Redmine

## What is Redmine?

If you search for a free project management tool most likely you will end up with Redmine. This is an open source Ruby on Rails web application, which can be considered to be the de facto flagship of project management solutions in the open source world. It supports all you need for effective project management: member roles, permissions management based on roles, Gantt charts, scheduling, calendar, roadmap, versions management, documents management, news delivery, files directory, activity view, and more. With third-party extensions, you may also get invoice management, Scrum backlogs, Kanban board, burndown charts, and more. But it's not only a matter of project management.

No one can tell whether Redmine is more a project management tool or an issue tracker. Ideally, a good issue tracker must have some project management features. In Redmine these two components are combined flawlessly. But what makes it a perfect issue tracking application is the fully-configurable workflow, which lets defining issue status change permissions for each role and tracker (issue type). As an issue tracker, Redmine also supports such essential features as priorities, sub issues, watching, comments, custom fields, listing filters, and more. Anyone who has ever worked in a team understands the importance of project documentation. For this purpose, many teams establish Wiki sites. Redmine ships with its own per-project Wiki system supporting the Textile markup syntax and source code syntax highlighting. The staggering thing is that this Wiki syntax is supported all over the Redmine that is in issue descriptions, comments, news, and so on. The syntax also allows you to have cross links to other issues and projects.

For public projects support, Redmine comes with a simple bulletin board module, which allows you to have as many forums as you need. Each forum can have an arbitrary number of threads. Forums and threads can be also watched.

Having read the above paragraphs one may think that the only missing feature for making Redmine host one's projects is some support for version control systems. But such support is also available. Redmine can be used as a repository source code browser. The repository component also integrates flawlessly into other Redmine components such as issue tracker and Wiki. For example, an issue can have list of revisions related to it, a Wiki page can link to a revision, a commit, a source file, and more. With some additional extensions, Redmine also can be turned into a repository manager or even play like Github. The list of supported version control systems is also impressive. They are **Subversion (SVN)**, Git, CVS, Mercurial, Bazaar, and Darcs.

All the things mentioned above make Redmine a perfect application for project hosting. Many individuals and organizations use it for this purpose. But it's not limited to a single project – it is multiproject and each project can have subprojects to any nesting level. Many companies utilize Redmine's collaborative capabilities for forge or labs sites. However, it is not even limited to software development. Other companies use Redmine for customer support, order fulfillment, task management, document management, and more.

I cannot describe Redmine without mentioning the people who created this fabulous software. As soon as you open Redmine, you see the name of its primary author at the bottom of each page (near the copyright section), Jean-Philippe Lang. A huge contribution to Redmine was also made by Eric Davis who then forked Redmine into ChiliProject.

## Why does Redmine succeed

The previous topic makes Redmine sound like some all-in-one software. To some extent it is. It was the evolution of Redmine that made it look like this. It is extremely popular these days and that's why it constantly gets new features. But what makes it so popular?

Having first seen Redmine I got the impression that it was a very easy-to-use and friendly application. It is not overloaded with design and UI elements and everything seems to be in its place. This makes users get used to Redmine and like it on first sight. The very first time I saw Redmine I also thought that perhaps it was too limited for my needs mainly because it looked too simple. Both impressions were wrong; it's not an easy-to-use software and it's not limited. The easiness of the look and feel however does its job. If you need only basic features you are ready to use Redmine right after you have seen it for the first time. When you need more advanced features you need to spend some time learning them. That's the main thing that makes Redmine so popular I believe.

The right tools are built with the right technologies. What make Redmine so "right" are Ruby and Rails. **Ruby** is known to be perhaps the most modern metaprogramming truly object-oriented language. This programming language is very flexible and is considered to allow building powerful applications fast and easily. All this can be said about Redmine. This all can be said about Ruby-on-Rails as well. **Rails** is a web framework similar to Symphony and Zend Framework but unlike the others it is the de facto standard for Ruby. Treat Rails as a construction set for building web services like Redmine. The names are so closely associated that many people believe they are about the same language. Ruby-on-Rails also became the source of inspiration for many frameworks and libraries such as CakePHP and Grails. Redmine is built on these technologies and this is the thing that makes it so good.

Of course, you can wonder what exactly is in Ruby-on-Rails that makes it good? The first thing is that Ruby (and therefore Ruby-on-Rails) supports metaprogramming. It's the technique which allows an application to modify its code (that is itself) at runtime. This means that there is almost nothing in Redmine that cannot be altered programmatically. Usually API of an application is limited to some functionality but there are no limitations in Ruby thanks to metaprogramming. What a good feature for plugin API, isn't it? The second thing is that Ruby-on-Rails establishes plugin API, which is used to develop Rails plugins called **engines**. So as you see Redmine does not actually need to provide plugin API to be extendable but it does. Redmine plugin API is built on top of the Rails engine API.

Thereby we come to the next thing, which makes Redmine so popular – its plugins. If you are familiar with Ruby and Ruby-on-Rails, you need to learn a little to start developing Redmine plugins. Taking into account that Ruby-on-Rails is very popular nowadays, Redmine has a huge amount of potential developers. Therefore it has a large variety of plugins. With its plugins you can even turn Redmine into a CRM or help desk. Some of the plugins will be reviewed in *Chapter 10, Plugins and Themes*.



The known issue is (at least partial) Redmine versions incompatibility. Redmine plugin API used to be changed from version to version without good backwards compatibility. It's even more critical as many plugins use metaprogramming for altering non-API (core) functionality. But in fact it's impossible to preserve full backwards compatibility in such cases. The same compatibility issue affects the Rails API that is especially seen in Redmine 2.0, which switches from Rails 2 to Rails 3. Hence, when selecting a plugin you should always check if it is compatible with the Redmine version you are using.

And last but not the least important benefits are that Redmine is cross-platform open source and freely available. Open source code and GPL license makes any modification possible. Nothing limits you in making Redmine better fit your needs.

## Walk through the Redmine interface

It's always better to meet than to only hear about. I cannot imagine a reader who is familiar with Redmine but has never seen it. So let's start with checking the Redmine interface.

As mentioned earlier, Redmine has an easy-to-use and simple interface. The following screenshot shows its **Home** page:

The screenshot shows the Redmine interface for the Packt Publishing project. At the top, there is a navigation bar with links for Home, My page, Projects, Administration, and Help. On the right side of the navigation bar, it indicates the user is logged in as 's-andy' and provides links for My account and Sign out. Below the navigation bar is a search box. The main content area is titled 'Home' and contains a site introduction for Packt Publishing, a 'Latest projects' section listing 'Mastering Redmine' (dated 05/21/2012 02:53 pm), and a 'Latest news' section with an announcement about the book's target audience and author information. The footer of the page states 'Powered by Redmine © 2006-2012 Jean-Philippe Lang'.

On the top-left of the page, we have the global menu (**Home**, **My page**, and so on). To the right-hand side of the page, we have the account menu (**Logged in as**). The blue area below these menus contains the site title. And finally the content area contains site introduction, recent news for all projects and recent projects.

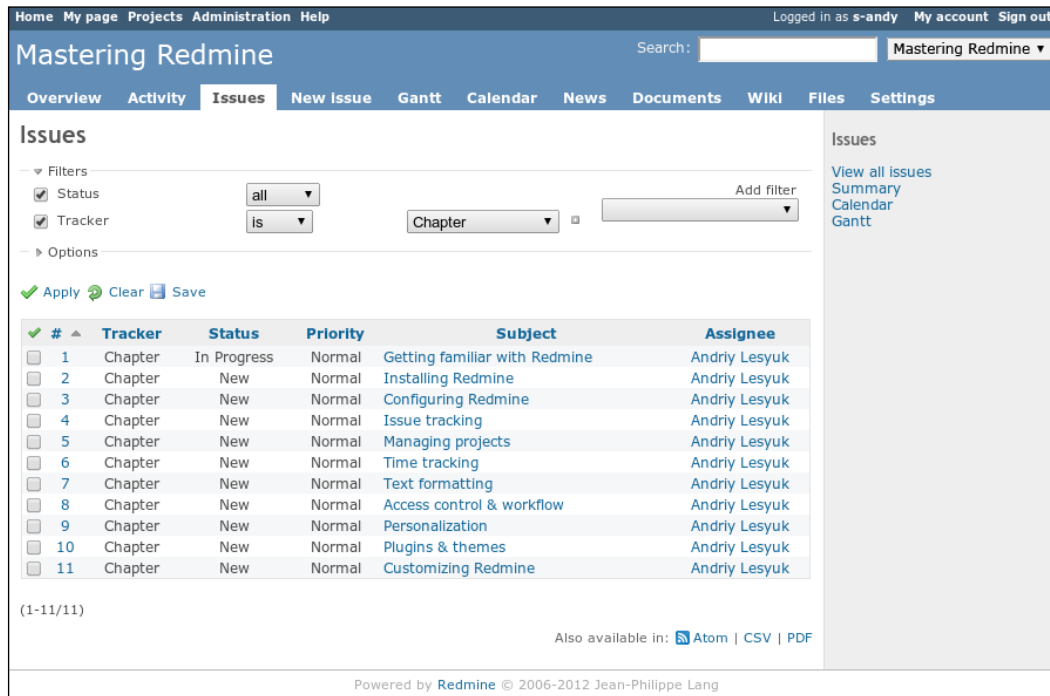
Actually any page in Redmine can be either a global page (the previous screenshot) or a project page:

The screenshot shows the Redmine interface for the 'Mastering Redmine' project. The top navigation bar is similar to the previous screenshot, but the search box now contains 'Mastering Redmine'. Below the navigation bar, there is a sub-navigation bar with tabs for Overview, Activity, Issues, New Issue, Gantt, Calendar, News, Documents, Wiki, Files, and Settings. The main content area is titled 'Overview' and contains a project description, a 'Members' section listing the Acquisition Editor (Wilson D'souza), Project Coordinator (Abhishek Kori), and Author (Andriy Lesyuk), an 'Issue tracking' section showing 11 open issues across various categories, and a 'Latest news' section with an announcement about the book's target audience. On the right side, there is a 'Spent time' section showing 0.00 hours and a 'Details | Report' link. The footer of the page states 'Powered by Redmine © 2006-2012 Jean-Philippe Lang'.

## Getting Familiar with Redmine

This is the start page for a project or the project home page. Most of the time when working with Redmine, you will be interacting with project pages. The blue top area on a project page contains the project title. The project menu is displayed below the title. The project home page contains a description of the project, issues summary, members summary, and the latest news about the project. Most project pages also have the sidebar with contextual information, such as links and forms.

As one of the primary features of Redmine is issue tracking, let's check the issue list:



The screenshot shows the Redmine interface for the 'Mastering Redmine' project. The top navigation bar includes 'Home', 'My page', 'Projects', 'Administration', and 'Help'. The user is logged in as 's-andy'. The main navigation menu includes 'Overview', 'Activity', 'Issues', 'New Issue', 'Gantt', 'Calendar', 'News', 'Documents', 'Wiki', 'Files', and 'Settings'. The 'Issues' section is active, showing a list of 11 issues. The table has columns for '#', 'Tracker', 'Status', 'Priority', 'Subject', and 'Assignee'. The issues are all assigned to 'Andriy Lesyuk'. The first issue is 'Getting familiar with Redmine' with a status of 'In Progress'. The other 10 issues are 'New' and cover topics like 'Installing Redmine', 'Configuring Redmine', 'Issue tracking', 'Managing projects', 'Time tracking', 'Text formatting', 'Access control & workflow', 'Personalization', 'Plugins & themes', and 'Customizing Redmine'. The page also includes a sidebar with links to 'View all Issues', 'Summary', 'Calendar', and 'Gantt'. The footer indicates the page is powered by Redmine © 2006-2012 Jean-Philippe Lang.

#	Tracker	Status	Priority	Subject	Assignee
1	Chapter	In Progress	Normal	Getting familiar with Redmine	Andriy Lesyuk
2	Chapter	New	Normal	Installing Redmine	Andriy Lesyuk
3	Chapter	New	Normal	Configuring Redmine	Andriy Lesyuk
4	Chapter	New	Normal	Issue tracking	Andriy Lesyuk
5	Chapter	New	Normal	Managing projects	Andriy Lesyuk
6	Chapter	New	Normal	Time tracking	Andriy Lesyuk
7	Chapter	New	Normal	Text formatting	Andriy Lesyuk
8	Chapter	New	Normal	Access control & workflow	Andriy Lesyuk
9	Chapter	New	Normal	Personalization	Andriy Lesyuk
10	Chapter	New	Normal	Plugins & themes	Andriy Lesyuk
11	Chapter	New	Normal	Customizing Redmine	Andriy Lesyuk

As you see the layout is quite simple and friendly. Collapsible boxes such as **Filters** and **Options** are used all over Redmine to hide rarely used elements. The order of issues in the issues-listing table can be changed by clicking on a header.

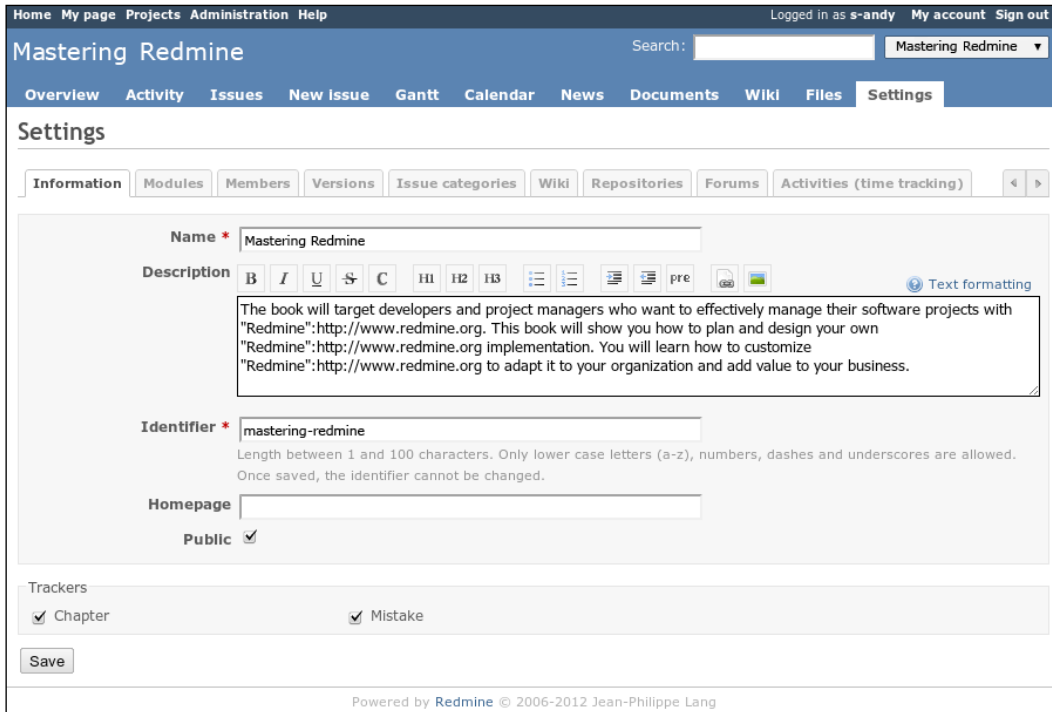
The following screenshot shows how an issue page looks:

The screenshot shows a Redmine issue page for 'Chapter #1'. The page is titled 'Mastering Redmine' and is logged in as 's-andy'. The issue is 'Getting familiar with Redmine', added by 'Andriy Lesyuk' 17 minutes ago. It is currently 'In Progress' with a 'Normal' priority. The 'Due date' is '06/12/2012' and the '% Done' is 80%. The issue is assigned to 'Andriy Lesyuk'. The description states that the chapter answers basic questions about Redmine and provides a component review. The page also shows a 'History' section with updates from 'Andriy Lesyuk' less than a minute ago, including changes to the description, due date, start date, and % Done. The page is powered by Redmine and includes a contextual menu for actions like Update, Log time, Watch, Copy, and Delete.

In the top-right corner of the content area we see the contextual menu, which is duplicated at the bottom of the page. A similar type of menu is used for other Redmine *objects* such as projects, Wiki pages, and more. The big yellowish box contains issue details. Below this box, there is the changes' history.

## Getting Familiar with Redmine

Another UI element, which is intensively used in Redmine is the tabular menu. Such a menu can be found, for example, in the project settings:



The screenshot shows the Redmine interface for the project 'Mastering Redmine'. The top navigation bar includes 'Home', 'My page', 'Projects', 'Administration', and 'Help'. The user is logged in as 's-andy'. The main navigation menu is active on 'Settings'. The 'Settings' page has several tabs: 'Information', 'Modules', 'Members', 'Versions', 'Issue categories', 'Wiki', 'Repositories', 'Forums', and 'Activities (time tracking)'. The 'Information' tab is selected, showing the following fields:

- Name \***: Mastering Redmine
- Description**: A text area with a rich text editor toolbar (B, I, U, S, C, H1, H2, H3, list, pre, image) and a 'Text formatting' link. The content is: "The book will target developers and project managers who want to effectively manage their software projects with "Redmine":http://www.redmine.org. This book will show you how to plan and design your own "Redmine":http://www.redmine.org implementation. You will learn how to customize "Redmine":http://www.redmine.org to adapt it to your organization and add value to your business."
- Identifier \***: mastering-redmine. A note below states: "Length between 1 and 100 characters. Only lower case letters (a-z), numbers, dashes and underscores are allowed. Once saved, the identifier cannot be changed."
- Homepage**: An empty text input field.
- Public**:

Below these fields is a 'Trackers' section with two checked items: 'Chapter' and 'Mistake'. A 'Save' button is at the bottom left. The footer reads: 'Powered by Redmine © 2006-2012 Jean-Philippe Lang'.

Here we also see another UI element, which is widely used in Redmine - the Wiki syntax enabled text area.

The following screenshot shows how administration settings are implemented in a similar way:

The screenshot displays the administration settings for a project named "Packt Publishing". The interface includes a top navigation bar with links for Home, My page, Projects, Administration, and Help. The user is logged in as "s-andy" and can access their account or sign out. A search bar and a "Jump to a project..." dropdown are also present.

The main content area is titled "Settings" and features several tabs: General, Display, Authentication, Projects, Issue tracking, Email notifications, and Incoming. The "General" tab is currently selected.

The settings are organized into a form with the following fields and options:

- Application title:** Packt Publishing
- Welcome text:** A rich text editor with a toolbar (Bold, Italic, Underline, Strikethrough, Code, H1, H2, H3) and a "Text formatting" button. The text reads: "Packt is a modern publishing company, producing cutting-edge books and eBooks for communities of developers, administrators, and newbies alike. Packt, pronounced Packed, published its first book Mastering phpMyAdmin for Effective MySQL Management in April 2004 and has".
- Maximum attachment size:** 5120 kB
- Objects per page options:** 25,50,100 (Note: Multiple values allowed (comma separated).)
- Days displayed on project activity:** 30 days
- Host name and path:** localhost:3000 (Example: 192.168.14.120)
- Protocol:** HTTP
- Text formatting:** textile
- Cache formatted text:**
- Wiki history compression:** none
- Maximum number of items in Atom feeds:** 15
- Maximum size of text files displayed inline:** 512 kB
- Maximum number of diff lines displayed:** 1500
- Attachments and repositories encodings:** (Note: Multiple values allowed (comma separated).)

A "Save" button is located at the bottom left of the settings form.

The right sidebar, titled "Administration", contains a list of navigation links: Projects, Users, Groups, Roles and permissions, Trackers, Issue statuses, Workflow, Custom fields, Enumerations, Settings (highlighted), LDAP authentication, Plugins, and Information.

At the bottom of the page, it states: "Powered by Redmine © 2006-2012 Jean-Philippe Lang".

This is another global page (similar to **Home**). Administration sections are available on the sidebar. Currently, we are in the **Settings** section, which also utilizes a tabular menu. On the **General** tab of the **Settings** section, we see Wiki-enabled **Welcome text** field.

This was a short walk through the Redmine interface to let you get used to it and understand its basic concepts. I hope you see that it is easy to use. We will review the interface in detail in subsequent chapters.

## MySQL, PostgreSQL, or SQLite

Redmine can be used with MySQL, PostgreSQL, or SQLite database backends (actually the same backends that are natively supported by Ruby-on-Rails). But which of those should you use? It's perhaps too important a question to be ignored in this book. No, I'm not going to praise one of the databases or criticize another (it's out of scope here) but I'll cover things you should consider when choosing a database backend.

In its *Installation Guide* in the Database section (<http://www.redmine.org/projects/redmine/wiki/RedmineInstall#Database>) at the time of writing this book, authors of Redmine added (recommended) to the right of **MySQL**. Let's try to understand why they recommended this backend.

Having worked with Redmine for a long time I noticed that most users use it with MySQL. It does not matter why they choose it but this means that Redmine has been tested better with this database backend. As long as plugins may use SQL directly (that is without using Rails SQL query builder) this argument becomes even more important. Also I'm sure there are plugins which have been never tested with other database backends.

If you are seeking a reliable source on what database is faster with Redmine, you can check Redmine build logs at <http://www.redmine.org/builds/index.html>. These logs contain results of tests for different versions of Ruby and different databases. Thus at the moment of writing this topic the latest revision r9739 in the latest stable branch 2.0-stable contained the following timings (showing only ruby-1.9.3-p194):

---

	mysql	postgresql	sqlite3
Tests run 1	304.131903s	320.018676s	276.440358s
Tests run 2	308.206684s	311.683513s	283.817843s
Tests run 3	112.832042s	105.113524s	94.457021s
Total	725.170629s	736.815713s	654.715222s

---

According to these data, SQLite is fastest and PostgreSQL is slowest. You can check new data or take several revisions into account to make sure.

You can ask: Why not SQLite then, if it's the fastest? SQLite has scalability issues (you can't put it on a separate server) and can potentially have problems with concurrent access. So, MySQL looks to be the optimal backend.

## Textile or Markdown?

Good readability helps achieving perception. Rich formatting is very important for issue tracking software as it allows you to distinguish and highlight more important things thus drawing special attention to them. In Redmine, rich formatting is done using **Textile** syntaxes, which is supported almost in every text area. In addition, Redmine provides formatter API, which means that the default Textile formatter can be replaced.

There were attempts to add different syntaxes to Redmine Wiki but the only successful and still active non-native Wiki syntax plugin is the Markdown Redcarpet plugin ([https://github.com/aluminium/redmine\\_redcarpet\\_formatter](https://github.com/aluminium/redmine_redcarpet_formatter)). This plugin replaced the default Redmine formatter with the Markdown formatter, which uses the Redcarpet library. This is the same library that is used by Github. Therefore the syntax provided with the Redcarpet plugin is compatible with Github's **Markdown** syntax.

If you asked what formatter I recommend, I would answer *Textile*. The Textile formatter is native, well supported, and well tested. However, when choosing a formatter you should consider your audience; if your audience is mostly Github users you should perhaps try Markdown but still remember that the majority of Redmine installations use Textile and users may expect your Redmine to use Textile as well.

But let's not be too verbose and compare basic rules of these syntaxes:

---

	<b>Textile</b>	<b>Markdown</b>
<b>Bold text</b>	<code>*Bold*</code>	<code>**Bold**</code>
<b>Italic text</b>	<code>_Italic_</code>	<code>*Italic*</code>
<b>Underline text</b>	<code>+Underline+</code>	<code>&lt;ins&gt;Underline&lt;/ins&gt;</code>
<b>Inline code</b>	<code>@inline code@</code>	<code>`inline code`</code>
<b>Preformatted text</b>	<code>&lt;pre&gt;</code> .... <code>&lt;/pre&gt;</code>	Text which starts with spaces
<b>Syntax highlighting</b>	<code>&lt;pre&gt;&lt;code</code> <code>class="ruby"&gt;</code> ... <code>&lt;/code&gt;&lt;/pre&gt;</code>	<code>```ruby</code> ... <code>```</code>
<b>Bullet List</b>	<code>* Item 1</code> <code>* Item 2</code>	<code>* Item 1</code> <code>* Item 2</code>
<b>Numbered list</b>	<code># Item 1</code> <code># Item 2</code>	<code>1. Item 1</code> <code>2. Item 2</code>
<b>Headings</b>	<code>h1. Heading 1</code> <code>h2. Heading 2</code> ... <code>h6. Heading 6</code>	<code># Heading 1</code> <code>## Heading 2</code> ... <code>##### Heading 6</code>
<b>Links</b> "Anchor":http://link	<code>[Anchor] (http://link)</code>	
<b>Images</b>	<code>!image_url (Title)!</code>	<code>![Title] (image_url)</code>
<b>Tables</b>	<code> _ .Table _ .Heading </code> <code> Cell  Cell  </code>	<code> Table Heading </code> <code> ----- ----- </code> <code> Cell  Cell  </code>

---

## Selecting a Source Control Management (SCM)

Source Control Management (SCM) is better known as Revision Control Management. Revision control systems currently supported by Redmine include Subversion (SVN), Git, Mercurial, Bazaar, Darcs and CVS.

As mentioned, Redmine integrates SCMs nicely using them not just as source code readers but making more use of them. In Redmine, we can do the following: Associate a revision with issues (having revisions listed on issue pages). Automatically close an issue and/or change its done ratio when an appropriate commit has been made. Use Wiki syntax in commit messages. Refer a revision, a commit, or a file from a Wiki syntax powered content, which can be a Wiki page, an issue description, a project description, or a forum message and much more

But all this makes Redmine a source code browser not a SCM manager (that can be done with plugins however). So why is choosing the right SCM so important?

Unless you are fine with having SCM and Redmine integrated visually most likely you will want a deeper integration when you know the options. Three levels of Redmine and SCM integration can be distinguished as follows:

- Redmine as a source code browser
- Redmine as a SCM authenticator
- Redmine as a SCM manager

The basic browser level requires SCM clients to be installed on the same server Redmine is running on so you may want to use only some of the supported SCMs. Also the basic support of SCMs is not equal: revision graph (similar to one on Github) is supported only for Git and Mercurial, however, these two SCMs can be only local (*repositories should be on the same server, where the Redmine is*).

Revisions				Revision: <input type="text" value="master"/> <input type="button" value="OK"/>
#	Date	Author	Comment	
<a href="#">fb7abea9</a>	03 May 2012 15:41	José Valim	Merge pull request #6139 from pwim/extract_options-actionpack use extract_options!	
<a href="#">54174b5f</a>	03 May 2012 15:26	José Valim	Merge pull request #6141 from mhfs/mailler_tweaks Minor ActionMailer tweaks	
<a href="#">e821611c</a>	03 May 2012 15:21	Paul McMahon	use extract_options!	
<a href="#">5f2f9b57</a>	03 May 2012 15:11	Marcelo Silveira	No need to force conversion to Symbol since case ensures it's already one.	
<a href="#">3d021951</a>	03 May 2012 15:10	Marcelo Silveira	No need to work around 1.8 warnings anymore.	
<a href="#">ab7b5501</a>	03 May 2012 14:59	Vijay Dev	Merge pull request #6137 from FLOChip/cache_documentation cache_store has an extra option of :null_store.	
<a href="#">e608588d</a>	03 May 2012 14:58	Oscar Del Ben	Update command line guide	
<a href="#">5c0cb3d</a>	03 May 2012 11:50	José Valim	Merge pull request #6138 from bogdan/routes RouteSet: remove some code dups	
<a href="#">7273adab</a>	03 May 2012 11:28	Teng Siong Ong	cache_store has an extra option of :null_store.	
<a href="#">c9e809c8</a>	03 May 2012 11:20	Paul McMahon	I found it strange that this guide is redirecting questions to a specific person. Heiko Webers' (@hawe) last blog post is a year and a half old, so it's not obvious that he's still active with Rails security. If he is, feel free to revert.	
<a href="#">3e541799</a>	03 May 2012 10:53	Jon Leighton	Merge pull request #6134 from carlosantoniodasilva/ar-relation-private-methods Fix issue with private kernel methods and collection associations	
<a href="#">5d26c8f0</a>	03 May 2012 05:23	Carlos Antonio da Silva	Fix issue with private kernel methods and collection associations. Closes #2508 Change CollectionProxy#method_missing to use scoped.public_send, to avoid a problem described in issue #2508 when trying to use class methods with names like "open", that clash with private kernel methods....	

Redmine comes with `Redmine.pm`—a Perl module for Apache web server—which can be used to authenticate Subversion and Git users against Redmine. If the `Redmine.pm` tool has been integrated, you can control who has access to project Subversion/Git repository and what kind of access (read and/or write) they have by simply managing project members (and roles). Something similar (with additional changes made to Redmine and/or system) can be done for Mercurial and Bazaar. Nevertheless Subversion and Git SCMs are best supported by `Redmine.pm` and their support is out of the box. Most likely you will have problems configuring other SCMs to authenticate against Redmine.

The only missing functionality for Redmine to become a full-featured SCM manager is the ability to create repositories. And such functionality also comes with Redmine and is provided by the `reposman.rb` command-line tool. This tool supports Subversion, Darcs, Mercurial, Bazaar, and Git (all except CVS). The problem is that in order for it to work you need to create a cron job executing it periodically. Luckily alternative solutions exist and are provided by third-party plugins such as SCM Creator, Redmine Git Hosting, and Redmine Gitis.

When choosing SCM, you should also consider your requirements, your experience, preferences of your team or audience, easiness or complexity of use and so on. However, the best integratable SCMs seem to be Subversion and Git.

## Selecting a web server and an application server

Redmine as a Ruby-on-Rails web application should be run under a web server. This can be an independent web server such as Apache, Nginx, or Lighttpd running Ruby-on-Rails using either Passenger, FastCGI, or a dedicated Ruby web server such as Unicorn or Thin.

A big options list, isn't it? Actually these are not all the possible options. Redmine can also be used with JRuby under a Java virtual machine, can be run under standalone Passenger, under Mongrel, under WEBrick, and more. But the above options were *chosen* by practical use and, therefore, are most commonly used. So we review only them here. They can be divided into three categories:

- Dedicated Ruby web server
- Dedicated Ruby web server plus for example, Nginx as a load balancer
- Separate web server with for example, Passenger module