HZ BOOKS

Apress®

借助开源Ruby程序库更好更快捷地开发应用程序

Ruby Gems 开发实战

Practical Ruby Gems

（美） David Berube 著

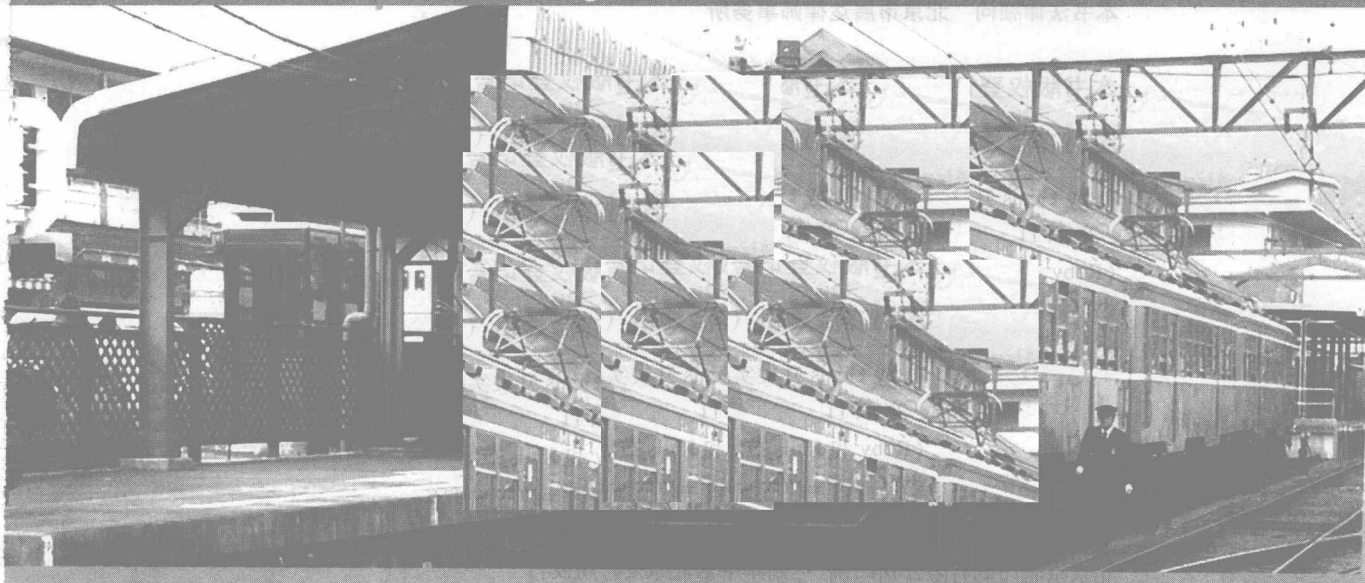
王 磊 寇晓丽 张建科 译

Ruby和Rails技术系列

机械工业出版社
China Machine Press

Ruby Gems 开发实战

Practical Ruby Gems



(美) David Berube 著
王 磊 寇晓丽 张建科 译

机械工业出版社
China Machine Press

本书是讨论如何在 Ruby 或者 Rails 项目中使用现有 Ruby 解决方案的全面指导书，同时它也涵盖了如何创建或者分发 Ruby 软件包的相关内容。本书分为三部分，分别为使用 RubyGems、使用特殊的 Ruby Gem 包和创建 Ruby Gem 包。

本书非常适合 Ruby 开发者阅读，也适合那些使用 Rails，或是希望对现有 Ruby、Rails 项目进行扩展的开发者。

David Berube：*Practical Ruby Gems*（ISBN：1-59059-811-3）.

本书版权登记号：图字：01-2008-1861

图书在版编目（CIP）数据

Ruby Gems 开发实战 /（美）贝鲁比（Berube，D.）著；王磊等译．—北京：机械工业出版社，2008.11

（Ruby 和 Rails 技术系列）

书名原文：Practical Ruby Gems

ISBN 978-7-111-24941-2

Ⅰ.R…　Ⅱ.①贝…　②王…　Ⅲ.计算机网络—程序设计　Ⅳ.TP393.09

中国版本图书馆 CIP 数据核字（2008）第 126982 号

机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码　100037）

责任编辑：周茂辉

北京京北印刷有限公司印刷 · 新华书店北京发行所发行

2008 年 12 月第 1 版第 1 次印刷

186mm × 240mm · 14.75 印张

标准书号：978-7-111-24941-2

定价：35.00 元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

译 者 序

Ruby，一种功能强大的、真正的解释型面向对象脚本语言，在它诞生的很长一段时间里，一直处于被大多数人遗忘的角落里。直到最近几年，当 David Heinemeier Hansson——Rails 的创造者在 2005 年“全球开源大会”（OSCON）被评为“年度最佳黑客”；当 Rails 赢得 2006 年度 Jolt“最佳 Web 开发工具”大奖；当《Agile Web Development with Rails》赢得 2006 年 Jolt“最佳技术图书”大奖；当众多闪耀的光环笼罩着这个新生的宠儿时，越来越多的人开始通过 Rails 认识 Ruby。

不可否认，Rails 真的是很好、很强大。当你认认真真地看完任何一本 Rails 书籍，就应该能使用 Rails 开发出一个不错的 Web 应用程序，但你是否真正明白 Ruby 和 Rails 的关系呢？也许你会毫不犹豫地说：“Rails 就是使用 Ruby 语言实现的一种 Web 快速开发框架”。从广义上讲没错，但这样回答好像少了些细节的东西。首先，Ruby 不是为 Rails 而生，但 Ruby 成就了 Rails——因为 Ruby 的简洁性、高效性，David Heinemeier Hansson 才使用它创造了一个优秀的 Web 框架，也就是后来大家所熟知的 Rails。因为 Rails 是由 Ruby 语言实现的，所以只要编写 Rails 应用程序，就必然离不开 Ruby。其次，为什么我们在一台安装了 Ruby 的计算机上，只用一条命令就能完成 Rails 的安装，为什么在 Rails 更新如此之快的今天，我们只用一条命令就能将已安装的 Rails 更新为最新发布的版本，这都要归功于 RubyGems——虽然它现在听起来很陌生。因为 RubyGems 的存在，使得我们在 Ruby 的环境中，仅仅使用一条命令就能完成 Rails 的下载、安装以及升级；因为 RubyGems 的存在，Rails 不仅仅是一个基于 Ruby 语言的优秀框架，更是一个能被 Ruby 开发者轻松管理和使用的 Ruby 库。那到底什么是 RubyGems 呢？

RubyGems 实际上是一个 Ruby 应用程序的管理器，它定义了一种通用的 Ruby 软件包格式——一个经过包装的 Ruby 应用程序或 Ruby 库，我们称之为 Gem，同时也为我们提供了一种简单、方便的方式来管理这些 Gem。通过使用 RubyGems，我们能基于标准、统一的接口构建并分发 Ruby Gem；通过使用 RubyGems，我们能搭建统一的 Ruby 软件存储仓库，供全世界的 Ruby 爱好者互相交流，共享优秀的 Ruby Gem；通过使用 RubyGems，我们能从 Ruby 软件存储仓库上查询、搜索、下载、安装合适的 Ruby Gem；通过使用 RubyGems，我们能方便地对本地安装的 Ruby Gem 进行升级或者删除；通过使用 RubyGems，我们能轻松地管理某 Ruby Gem 的多个版本，能处理不同 Ruby Gem 间的依赖关系；通过使用 RubyGems，我们也能完成其他工作；总之，RubyGems 是一个功能强大的 Ruby Gem 包管理平台。说了这么多，也许你还是对这些绕口的文字半信半疑，别着急，由于 Rails 也是一个 Gem，因此我们通过大家熟悉的 Rails，来真正领略一下 RubyGems 的强大之处：下载并安装 Rails，只需执行命令 gem install rails；升级 Rails，只需执行命令 gem update rails；删除 Rails，只需执行命令 gem uninstall rails；除此之外，你还可以在一个系统上安装多个版本的 Rails。试想一下，如果没有 RubyGems，让你手动安装、更新或者删除依赖于几十个 Ruby 软件包的 Rails，那对开发者而言简直就是一场噩梦，相信在你兴致勃勃地安装完 Rails 后，再也无心使用它了！对，这就是 RubyGems 的伟大之处。实际上，RubyGems 的更伟大之处在于：借助于 RubyGems 提供的标准 Gem 接口，开源阵营搭建了 Ruby 的软件存储仓库——RubyForge，它为我们提供了一套数以

万计的 Ruby Gem 解决方案集合（Rails 当然是其中最著名的一个，类似地，还包括轻量级的 Web 开发框架 Camping，验证信用卡的 Creditcard，处理图片的 RMagick，解析 RSS Feeds 的 Feedtools 等）。通过使用 RubyForge 平台，全世界的 Ruby 开发者可以互相交流，互相分享；通过使用 RubyForge 平台，每一个 Ruby 开发者都能充分利用开源社区提供的优秀解决方案迅速解决遇到的问题，同时，所有的 Ruby 爱好者，也能将解决问题的方案封装成 Gem 包，提供给更多的开发者使用。

本书的作者 David Berube——一位在 Ruby 开发方面有着丰富经验的资深专家，正是从 RubyGems 的使用角度入手编写了本书。在阐述了什么是 RubyGems 以及如何使用 RubyGems 后，作者精挑细选了当今最具影响的 34 个 Ruby Gem 包——包括如何使用 Creditcard 验证信用卡的有效性；如何使用 RMagick 处理图片的特殊效果；如何使用 Mongrel 服务于 Web 应用程序；如何使用 Rubyscript2exe 将 Ruby 应用程序转换成可执行应用程序等，并通过大量示例展示了这些 Gem 的强大功能以及为开发者所带来的便利之处，同时就如何解决一些棘手或易忽视的问题分享了其经验与技巧。最后，作者还提到了如何创建并分发自己的 Gem 包。总之，这是一本理论与实践相结合、通俗易懂、能帮助 Ruby 或 Rails 开发者提高开发效率和分享各自成果（Gem 包）的好书。

最后，感谢机械工业出版社华章分社使译者有机会翻译这本好书，感谢编辑以及众多同事们在本书后期制作过程中付出的辛勤劳动。特别感谢本书作者 David Berube，为我们带来了这样一本值得阅读和学习的好书。虽然译者已为该书的翻译付出了不懈的努力，但一定还有不当之处，欢迎读者朋友积极指正。

Ruby 的朋友们，请好好享受这本书吧，相信你们一定会受益匪浅的！

译　者

2008 年 5 月 11 日

作者简介

David Berube 是一名 Ruby 开发者、培训家、作家以及演说家。之前，他主要使用 PHP、Perl、C++ 和 Visual Basic 等从事软件开发。从 2003 年起，David 开始使用 Ruby 和 Ruby on Rails 从事软件开发。之后，当他撰写的 Ruby 相关文章在《Dr Dobb's Journal》上发表后，David 开始成为一名 Ruby 语言的拥护者。

David 使用 Ruby 以及 Rails 开发了多个项目，其杰出的成果包括：使用 Ruby on Rails 实现了 Cool-Ruby. com（一个追踪 Ruby 最新发展动态的站点，更多细节请访问 http://coolruby. com）；与 thoughtbot（www. thoughtbot. com）合作，实现了美国宣传论文竞赛奖（Sermo's America's Top Doc contest）的平台；与 Casting Frontier 合作，为洛杉矶开发了数字点播服务（digital casting services）平台。另外，David 还开发了很多其他的 Ruby 项目，包括著名的禽流感疫情新闻平台（关于该平台的更多细节，请访问其相关站点 http://www. birdflubreakingnews. com/）。

David 的著作已经在超过 65 个国家出版发行，包括《Linux Magazine》、《Dr Dobb's Journal》以及《International PHP Magazine》等。同时，David 也经常在高校授课并公开发表演讲，包括“MySQL and You”，“Making Money with Open Source Software”等主题。

如果你对本书有任何建议或者问题，可以随时访问站点 http://berubeconsulting. com 或者通过 E-mail 地址 djberube@ berubeconsulting. com 与 David 联系。

致 谢

感谢我的父母、我的姐妹，如果没有他们，我无法想象本书该如何完成；同时，也感谢那些支持过我的朋友，特别是 Wayne Hammar 和 Matthew Gifford。

另外，感谢那些曾经和我合作过的同事和朋友，我从你们身上学到了很多东西，特别感谢 Terry Simkin、Ted Roche、Bill Sconce、Bruce Dawson、K. C. Singh 和 Joey Rubenstein，也感谢 Peter Cooper 介绍我写这本书。

最后，感谢编辑 Keir Thomas 和 Jason Gilmore，感谢技术顾问 Yan Pritzker、项目经理 Richard Dal Porto 以及发行顾问 Candace English。

目录

第三部分 创建 Ruby Gem 包

第一部分

使用RubyGems

本部分主要介绍了什么是RubyGems，以及如何使用RubyGems。

第1章　什么是RubyGems

简而言之，RubyGems能让你在任何安装Ruby的地方分发和安装Ruby应用程序。

具体地说，RubyGems是适用于Ruby应用程序的包管理器。它能让你在任何运行Ruby的机器上安装Ruby编写的应用程序（又称为Ruby软件包或Gem包）。另外，如果只想安装某个Ruby软件包的特定部分，RubyGems还能自动处理它们之间的依赖关系。如果只想安装某个特定版本的Gem包，RubyGems也能处理不同版本之间的依赖关系。RubyGems提供了一种简单、易用的方式来管理Ruby软件包。

在Ruby的世界里，存在着各种各样的Gem包供你选择。譬如说，你的Web应用程序需要支持用户上传照片，还需要调整照片的尺寸，而它们的大小各不相同。对于这些功能，你可以自己编码实现，但如果使用了RMagick Gem包（关于RMagick的更多细节，请参见本书第25章），情况可能变得更简单。你甚至只需要几行代码，就可以使用RMagick实现照片的裁剪、旋转、锐化等功能。

有时，你可能希望使用Ruby on Rails（一个功能强大的基于MVC的Ruby Web框架，关于Rails的更多细节，请参见本书第23章）开发Web应用程序，但实际上，Rails包括了很多其他的Gem包，如果没有RubyGems，安装这些Gem包可能会比较麻烦。现在有了RubyGems，只需一条命令，你就能轻松地完成它们的安装。

在本章中，我们主要介绍了什么是RubyGems，RubyGems的特点以及RubyGems和其他包管理器的区别。

1.1　为什么使用RubyGems

首先，使用RubyGems安装Ruby软件包非常容易。譬如说，Instiki（http://instiki.org/）是一种Wiki系统（内容管理系统）。如果你想安装Instiki Gem包，只需要在Linux/Mac的Shell或者Windows的命令行中执行如下命令：

```
gem install instiki
```

当然，前提是你必须安装了RubyGems。在接下来的第2章和第3章中，我们将详细介绍如何安装RubyGems。现在，你只需明白，使用RubyGems安装Gem包，一条命令就足够了，其他的RubyGems会帮你搞定。

更重要的一点，如果你的Web应用程序采用Ruby实现，当主机出现问题时，使用RubyGems，你能方便地在其他机器上迅速搭建Web应用程序运行的环境。

1.1.1　标准化的软件包格式

RubyGems容许你为Ruby应用程序定义Gemspec。Gemspec是Ruby或者YAML形式的元数据集，主要是用来为Gem包提供一些信息的，譬如应用程序的名称、版本、描述信息等。发布时，Gemspec和其他相关源文件一起压缩成一个以.gem结尾的文件，这个.gem文件遵守统一的格式。

如果将这个 .gem 文件上传到 RubyForge，我们就能够使用 RubyGems 在线安装它，当然也能通过传统的 FTP 或者 HTTP 等方式下载安装。另外，因为这个 .gem 文件包含了 Ruby 应用程序的相关信息，所以你能使用命令 gem list 查看该 Gem 包的详细信息，或者你也能在 RubyForge 中搜索类似功能的 Gem 包（关于更多 gem list 命令的细节，请参见第 3 章；关于构建 Gem 包的更多细节，请参见第 34 章和第 35 章）。

另外，如果希望将某个 Gem 包升级到最新版本，你根本不需要查看某些具体的文档来找出新版本的变化，Gemspec 提供的标准格式使 RubyGems 能轻松比较出这些变化。

1.1.2　访问 Ruby 软件包存储仓库

RubyGems 提供了一个令人兴奋的功能，它能够访问 RubyForge（一个 Ruby 软件包的主要存储仓库，关于 RubyForge 的更多信息，请访问 http://rubyforge.org）。如果没有 RubyForge，你必须自己去网上搜索、下载然后安装某个 Ruby 软件包，有时还需要处理该软件包和其他软件包的依赖关系。现在有了 RubyForge，RubyGems 能够自动替你完成这些繁琐的工作。

很多 Ruby 的爱好者都从这个 Ruby 存储仓库上安装 Gem 包，不过这并不是唯一的选择。你也能从其他的位置安装 Gem 包（后面会详细介绍如何搭建个人的 Gem 包服务器）。不同的操作系统，有不同的包管理器和集成的包存储仓库。当把某个 Ruby 应用程序从一个操作系统移植到另一个操作系统时，这些差异可能会给你带来麻烦。但有了 RubyGems，无论使用何种操作系统，只要你能够安装 RubyGems，就能够通过它访问 RubyForge。

1.1.3　使用 Gem 服务器重新分发 Gem 包

你可以在局域网或互联网上搭建自己的 Ruby Gem 服务器。譬如说，为了提高下载速率，你可以将小组内部使用的 Gem 包，都放到局域网的 Gem 服务器上。

另外，根据具体情况，你还可以选择从自己的站点或者其他 Gem 服务器下载 Ruby 软件包，而不是必须从 RubyForge 获取它们。关于这部分的更多细节，详见本书第 35 章。

1.1.4　处理软件包依赖关系

RubyGems 能够自动处理 Ruby 软件包之间的依赖关系。换句话说，当你安装一个 Gem 包时，RubyGems 能够检测到该包所依赖的其他 Gem 包，并会提示你是否需要安装。

事实上，大多数 Ruby 应用程序都会依赖于其他 Ruby 软件包，而其他 Ruby 软件包也可能会依赖于更多的包。如果没有 RubyGems，你可能需要花很长时间来理清这些包之间的依赖关系。但现在，你不必担心，RubyGems 会帮你搞定一切。

1.1.5　处理版本的依赖关系

RubyGems 能管理多个不同版本的 Ruby 软件包。譬如说，某个应用程序用到了 ActiveRecord（http://rubyforge.org/projects/activerecord/），而它却和当前环境中的 ActiveRecord 版本不同，这可能会导致某些问题；或者，当某个应用程序使用的 Gem 包版本较旧，换成新版本的包后，原程序却无法运行；或者，当应用程序使用了某个最新版本的 Gem 包所提供的功能，而旧版本的包完全不提供类似功能，此时使用老版本的 Gem 包，程序就无法正常运行。不过别担心，RubyGems 能帮

助你解决这些麻烦。关于这部分的更多细节，我们将在本书的第 4 章中详细介绍。

1.1.6　透明地替换 Ruby 库

使用 RubyGems，能方便地将 Gem 包与其对应的 Ruby 库互相替换。譬如说，你想使用 Camping（一个基于 Rails 的轻量级 Web 开发框架，更多关于 Camping 的细节，请参考本书第 7 章）开发 Web 应用程序。按照传统的库安装方式安装后，你可以这样在代码中引用它：

```
require 'camping'
```

如果删除以库方式安装的 Camping，使用 Gem 包重新安装后，你不需要对现有的代码做任何修改，仍然能继续使用 Camping 提供的所有功能，反之亦然。换句话说，你完全可以不用考虑代码中使用的 Camping 是以库的方式还是以 Gem 包的方式安装，它对应用程序完全透明。这一点，给我们带来了极大的好处。当发布应用程序时，用户不需要安装 RubyGems，也不需要安装应用程序用到的 Gem 包，只需要有相应的库存在即可。

另外，有一点需要注意，如果需要安装某个指定版本的 Gem 包，需要在代码中明确的指明（关于这部分细节，请参见本书第 4 章）。

1.1.7　处理操作系统的差异

对于任何类型的操作系统，只要它能运行 Ruby，就能运行 RubyGems，因此你能在任何运行 Ruby 的地方使用 RubyGems。而相比其他的包管理器，譬如 Windows 的 MSI、Debian Linux 的 Apt（Advanced Package Tool）、RedHat 的 Yum 以及 Mac OS 的 DarwinPorts，虽然它们也能简化 Ruby 应用程序的安装，但这些包管理器都是与操作系统紧密相关的，有很大的局限性。在接下来的 1.2 节中，我们将详细讨论 RubyGems 与其他包管理器的异同。

1.2　与其他包管理器的比较

Ruby 应用程序对所有人都是公平的。不管你是否喜欢 Ruby 这门语言，你都能轻松地使用它们。也正因为如此，大多数操作系统自带的包管理器，譬如 Apt 或者 Yum，也能安装 Ruby 软件包。不过，对于 Ruby 的爱好者，建议最好使用 RubyGems 来安装 Ruby 软件包。因为它能帮助你选择功能最完善或者最新版本的 Gem 包。另外，如果希望系统上能安装某个 Gem 包的多个不同版本，只有 RubyGems 能帮你实现，其他的包管理器目前望尘莫及。

另外，如果你使用其他的包管理器安装 Ruby 应用程序，不仅执行命令不同，可能多少还会存在些限制。譬如说，如果使用 RubyGems 的方式安装 MySQL 软件包，应该执行如下命令：

```
gem install mysql
```

而使用 Ubuntu Linux 的 apt-get 安装时，则执行的命令为：

```
apt-get install libmysql-ruby1.8
```

使用 OS X 下的 DarwinPorts 安装时，执行命令为：

```
port install rb-mysql
```

别忘了，必须是 root 用户才能执行这些命令，或者也可以使用 sudo，它能帮助你以 root 用户的权限执行该命令。

有时候，你能通过 Apt 或者其他的包管理器安装 Ruby 应用程序。但通常情况下，这些包管理器可选择的 Ruby 应用程序都比较有限，幸运的话，你也许正好能找到需要的。请参考 Linux 分发其他包管理器的文档。

不过有一点需要注意，尽管能够使用操作系统自带的包管理器安装 Ruby 应用程序包，但我们并不推荐这种方式。因为不同的包管理器，有不同的软件包存储仓库，从上面获取的 Ruby 软件包有可能不是最新的。而使用 RubyGems，则不用担心这个问题，它能自动为你安装最新版本的 Ruby 应用程序包。

有些读者可能会问，如果不使用任何包管理器，我能安装 Ruby 应用程序吗？答案是当然能！你可以先找到 Ruby 安装路径下的 lib 目录，然后使用脚本或直接将 Ruby 应用程序包复制到该目录下就可以了。不过使用这种方式，你得亲自去网上下载需要的 Ruby 应用程序包，并需要仔细阅读它的安装说明或注意事项。

第2章 安装RubyGems

为了能运行本书的示例，你必须先安装RubyGems。有了RubyGems，才能更容易地下载、安装和使用Ruby应用程序包，包括本书中提到的所有Ruby应用程序包。不过在使用RubyGems之前，你的机器上必须先安装Ruby。在本章中，我们将主要介绍如何安装Ruby、RubyGems以及如何对已经安装的RubyGems进行升级。

注意 如果已经安装了RubyGems，可以略过本章。

2.1 安装Ruby

在使用RubyGems之前，你必须先安装Ruby。默认情况下，Mac操作系统已经安装了Ruby，如果你使用的是Mac操作系统，可以略过此部分，直接阅读2.2节。另外，目前大多数Linux系列的操作系统，也都默认安装了Ruby。关于如何确定当前系统是否已经成功安装了Ruby，我们会在2.1.4节中详细介绍。Windows系列的操作系统，默认情况下不会安装Ruby，如果你的系统是Windows系列，请直接阅读2.1.3节。

2.1.1 你已经安装了Ruby

嗯？我记不清当前操作系统是否已经安装过Ruby了？没关系，如果忘记了是否安装过Ruby，只需要在Linux/Mac的Shell或Windows的命令行中执行如下命令：

```
ruby -v
```

如果返回结果为“command not found”，则表示当前系统没有安装Ruby。

如果得到如下的输出结果，则表示当前系统已经安装了Ruby。

```
ruby 1.8.4 (2006-04-14) [i386-mswin32]
```

在如上所示的输出结果中，紧跟在“ruby”后的几位数字1.8.4，表示当前Ruby的版本号。如果希望能正常使用RubyGems及其存储仓库上的大部分Ruby应用程序包，那你至少应该安装Ruby的1.8.4版本或更高版本。

2.1.2 在Linux上安装Ruby

在Linux平台上安装Ruby，目前主要有三种方式。一是使用Apt安装。一般地，在基于Debian系列的操作系统上，如Ubuntu等，使用的包管理器大部分都是Apt。二是使用Yum安装。Yum是基于RedHat系列操作系统上的包管理器，功能和Apt类似。在这些操作系统上，由于有包管理器的存在，安装Ruby相对比较简单。但如果操作系统不支持Apt或者Yum，那只能使用另外一种方式：编译源码安装。本节的最后，我们将详细讨论如何使用编译源码的方式安装Ruby。

在 Debian 上使用 Apt 安装 Ruby

Apt 是一个较受大家欢迎的包管理器，主要用于 Debian Linux 上或基于 Debian 系列的操作系统上，譬如 Ubuntu Linux、Lindows 和 Xandros 等。Apt 基本上类似于 RubyGems，能够提供基于命令行的软件包下载、安装和删除的功能。另外，Apt 也能够用于非 Debain 系列的操作系统，但这些操作系统并没有默认安装 Apt。

如果希望使用 Apt 安装 Ruby，只需在 Linux 的 Shell 中执行如下命令：

```
sudo apt-get install ruby*
```

Apt 会自动下载并安装 Ruby。安装成功了吗？接下来的 2.1.4 节将告诉你如何测试 Ruby 是否安装成功。

注意　命令 apt-get install ruby * 将安装所有名称以“ruby”开头的应用程序包，其中也包括安装 RubyGems 时用到的相关包。

在 Red Hat 上使用 Yum 安装 Ruby

Yum 类似于 Apt，也是一个优秀的包管理器。它适用于所有 Fedora 内核的 Linux 系统。如果希望使用 Yum 安装 Ruby，需要在 Linux 的 Shell 中执行如下命令：

```
yum install ruby
```

Yum 会自动下载并安装 Ruby。2.1.4 节将告诉你如何测试 Ruby 是否安装成功。

在 Linux 上进行 Ruby 编译源码安装

除了使用包管理器 Apt 或者 Yum 安装 Ruby，你还可以选择编译源码的方式安装。实际上，如果希望 Ruby 与操作系统具有较好的兼容性，编译源码安装是最好的解决方案。不过使用这种方式安装前，必须确保操作系统上已经安装了 gcc（关于如何安装 gcc，请根据不同的操作系统，查看具体的相关文档），然后下载并解压 Ruby 的源代码包（ftp://ftp.ruby-lang.org/pub/ruby），进入解压后的目录，在 Linux 的 Shell 中执行如下命令，即可完成 Ruby 的编译和安装。

```
./configure
make
make test
make install
```

安装完成后，可以测试 Ruby 是否安装成功（见第 2.1.4 节）。

2.1.3　在 Windows 上安装 Ruby

在 Windows 系列的操作系统上，一般使用 Ruby 一键安装包（One-Click Installer）来安装 Ruby。Ruby 一键安装包由 Ruby Central（http://rubycentral.org/）组织开发，是一个已编译过、能自解压的 Windows 二进制文件，你可以从 http://rubyinstaller.rubyforge.org/下载。目前，它仅能用于 Windows 系列的操作系统。

如同一键安装包的名字所示，使用它安装 Ruby 非常简单。你不需要启动任何程序，也不必在命令行中输入任何命令，双击该安装包，运行后，按照提示一步步操作，即可轻松完成安装。因为一键安装包比较符合 Windows 的传统安装方式，所以它受到很多人的青睐。安装完成后，2.1.4 节将告诉你如何测试 Ruby 是否安装成功。