**用 RPM 打包软件，第 1 部分**

**在 Red Hat Linux 7.1 上使用 RPM**

RPM 是广泛使用的用于交付 Linux 软件的工具，用户可以轻松的安装用 RPM 打包的产品。在本文中（系列文章的第 1 篇），IBM 软件工程师 Dan Poirier 向您演示如何在 Red Hat Linux 7.1 系统上用 RPM 打包简单软件。

1https://dw1.s81c.com/developerworks/i/v17/dw-cmts-arrow.png [评论](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/index.html#icomments)

[Dan Poirier](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/index.html#authorN10019) ([poirier@us.ibm.com](mailto:poirier@us.ibm.com?subject=%E7%94%A8%20RPM%20%E6%89%93%E5%8C%85%E8%BD%AF%E4%BB%B6%EF%BC%8C%E7%AC%AC%201%20%E9%83%A8%E5%88%86&cc=poirier@us.ibm.com)), 软件工程师, IBM

2001 年 11 月 01 日

* [expand](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/index.html#toggle)**内容**



在 IBM Bluemix 云平台上开发并部署您的下一个应用。

[**开始您的试用**](https://developer.ibm.com/sso/bmregistration?lang=zh_CN&ca=dwchina-_-bluemix-_-part1-_-sidebar)

RPM（Red Hat Package Manager）是用于 Linux 分发版（distribution）的最常见的软件包管理器。因为它允许分发已编译的软件，所以用户只用一个命令就可以安装软件。

RPM 是 Linux“标准基本库”版本 1.0.0 指定的安装工具。在 Linux 分发版前 10 名中，有 8 个是基于 RPM（请参阅“Comparison of Linux Distributions”，它位于本文后面的 [参考资料](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/index.html#resources)中列出的 distrowatch.com 上）。即使某些通常不使用 RPM 的分发版，如 Debian，也有可用工具将 RPM 转换成它们自己的格式。在 Linux 上，对于除开发人员以外的任何人，RPM 也是用来打包软件的最佳选择。

不论您是自己开发软件，还是通过提供代码以从中创建 RPM 软件包向开放源码项目提供帮助，本文都会帮助您入门。顺便要说的是，本系列的后续文章会涉及下列主题：构建 RPM 软件包而未必是 root 用户，在构建软件之前为其打补丁，安装和卸载时运行脚本以及在安装或卸载 *其它*软件包时运行脚本。

**简单实例**

我将从简单的实例开始，主要使用 RPM 缺省值。然后，我会添加上几个可选特性。

RPM 软件包从源文件形式的程序开始，作好了编译准备。我不想创建一个没有实际意义的示例，而是选择使用 GNU Indent 程序（请参阅 [参考资料](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/index.html#resources))。

在 Linux 下创建 Indent 非常容易。 indent-2.2.6.tar.gz 文件位于当前目录之中，所有您要做的就是：

**手工构建 indent**

$ tar xzf indent.2.2.6.tar.gz

$ cd indent-2.2.6

$ ./configure

$ make

$ make install

如果您已经构建过很多开放源码项目，那么这可能看起来很熟悉。 unpack；./configure；make；make install 序列是典型的使用 GNU 自动配置（autoconf）工具的软件。因为这太普通了，所以我在这里描述的关于 indent 的大部分东西，几乎不加改变就可以用于其它开放源码项目。

现在假设您正在使用 Red Hat 7.1。在本文的后面，我将给出一些 [在其它 Linux 分发版上使用 RPM 的建议](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/index.html#h17361)。

在 Red Hat 7.1 上， **在继续以前，请确保您已安装了 rpm-build 软件包。**为检查是否安装，请运行 rpm -q rpm-build 。您应该看到一些类似于 rpm-build-4.0.2-8 的信息（版本可能会不同）。如果您看到的是 package rpm-build is not installed ，则将需要从 Red Hat 安装 CD 上安装它。

**制作基本 RPM 软件包**

为了构建 RPM 软件包，您需要写一个名为 *spec* 文件的 RPM 输入文件，该文件告诉 RPM 如何构建和打包您的软件。编写 spec 文件您需要：

1. 创建文件 [indent-1.spec](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/indent-1.spec)，如下所示。您可以任意地给它命名并把它放到任何地方；RPM 对这些没有要求。
2. 以 root 用户登录。
3. 将 indent-2.2.6.tar.gz 文件复制到 /usr/src/redhat/SOURCES。
4. 运行 rpm -ba indent-1.spec ，将 indent-1.spec 改为您使用的名字。

**第一个 spec 文件：indent-1.spec**

Summary: GNU indent

Name: indent

Version: 2.2.6

Release: 1

Source0: %{name}-%{version}.tar.gz

License: GPL

Group: Development/Tools

%description

The GNU indent program reformats C code to any of a variety of

formatting standards, or you can define your own.

%prep

%setup -q

%build

./configure

make

%install

make install

%files

%defattr(-,root,root)

/usr/local/bin/indent

%doc /usr/local/info/indent.info

%doc %attr(0444,root,root) /usr/local/man/man1/indent.1

%doc COPYING AUTHORS README NEWS

您应该看到 RPM 解包这个 tar 文件，编译并安装它。 在 Red Hat 7.1 上，工作目录将是 /usr/src/redhat/BUILD。

最后，RPM 将创建两个 RPM 文件。将在 /usr/src/redhat/SRPMS/indent-2.2.6-1.src.rpm 中创建一个源 RPM 文件，而在 /usr/src/redhat/RPMS/i386/indent-2.2.6-1.i386.rpm 中创建一个二进制 RPM 文件。

源 RPM 文件简单地捆绑了 spec 文件和构建软件包用到的所有源文件和补丁文件。如果您选择分发它，则其他人可以很容易地用它重建您的软件。二进制 RPM 文件仅包含已编译的软件和如何安装的信息。

**RPM 做什么**

下面总结了在您运行 rpm -ba filename.spec 时，RPM 都做些什么：

* 读取并解析 filename.spec 文件
* 运行 %prep 部分来将源代码解包到一个临时目录，并应用所有的补丁程序。
* 运行 %build 部分来编译代码。
* 运行 %install 部分将代码安装到构建机器的目录中。
* 读取 %files 部分的文件列表，收集文件并创建二进制和源 RPM 文件。
* 运行 %clean 部分来除去临时构建目录。

**spec 文件的内容**

spec 文件有几个部分。第一部分是未标记的；其它部分以 %prep 和 %build 这样的行开始。

**头**  
第一部分（未标记）定义了多种信息，其格式类似电子邮件消息头。

Summary 是一行关于该软件包的描述。

Name 是该软件包的基名， Version 是该软件的版本号。 Release 是 RPM 本身的版本号 ― 如果修复了 spec 文件中的一个错误并发布了该软件同一版本的新 RPM，就应该增加发行版号。

License 应该给出一些许可术语（如：“GPL”、“Commercial”、“Shareware”）。

Group 标识软件类型；那些试图帮助人们管理 RPM 的程序通常按照组列出 RPM。您可以在 /usr/share/doc/rpm-4.0.2/GROUPS 文件看到一个 Red Hat 使用的组列表（假设您安装的 RPM 版本是 4.0.2）。但是您还可以使用那些组名以外的名称。

Source0 、 Source1 等等给这些源文件命名（通常为 tar.gz 文件）。 %{name} 和 %{version} 是 RPM 宏，它们扩展成为头中定义的 rpm 名称和版本。因此，在这个实例中， Source0 被设置为 indent-2.2.6.tar.gz 。

不要在 Source 语句中包含任何路径。缺省情况下，RPM 会在 /usr/src/redhat/SOURCES 中寻找文件。请将您的源文件复制或链接到那里。（要使 spec 文件尽量可移植的话，应当尽量避免嵌入自己开发机器上的假想路径。其他开发人员就可以指示 RPM 在别的目录下查找源文件，而不用修改您的 spec 文件。）

**描述**  
接下来的部分从 %description 行开始。您应该在这里提供该软件更多的描述，这样任何人使用 rpm -qi 查询您的软件包时都可以看到它。您可以解释这个软件包做什么，描述任何警告或附加的配置指令，等等。

**Shell 脚本**  
下面几部分是嵌入 spec 文件中的 shell 脚本。

%prep 负责对软件包解包。在最常见情况下，您只要用 %setup 宏即可，它会做适当的事情，在构建目录下解包源 tar 文件。加上 -q 项只是为了减少输出。

%build 应该编译软件包。该 shell 脚本从软件包的子目录下运行，在我们这个例子里是 indent-2.2.6 目录，因而这常常与运行 make 一样简单。

%install 在构建系统上安装软件包。这似乎和 make install 一样简单，但通常要复杂些。我将在下面解释这点。

**文件列表**  
%files 列出应该捆绑到 RPM 中的文件，并能够可选地设置许可权和其它信息。

在 %files 中，您可以使用 **一次**%defattr 来定义缺省的许可权、所有者和组；在这个示例中， %defattr(-,root,root) 会安装 root 用户拥有的所有文件，使用当 RPM 从构建系统捆绑它们时它们所具有的任何许可权。

可以用 %attr(permissions,user,group) 覆盖个别文件的所有者和许可权。

可以在 %files 中用一行包括多个文件。

可以通过在行中添加 %doc 或 %config 来标记文件。 %doc 告诉 RPM 这是一个文档文件，因此如果用户安装软件包时使用 --excludedocs，将不安装该文件。您也可以在 %doc 下不带路径列出文件名，RPM 会在构建目录下查找这些文件并在 RPM 文件中包括它们，并把它们安装到 /usr/share/doc/%{name}-%{version} 。以 %doc 的形式包括 README 和 ChangeLog 这样的文件是个好主意。

%config 告诉 RPM 这是一个配置文件。在升级时，RPM 将会试图避免用 RPM 打包的缺省配置文件覆盖用户仔细修改过的配置。

**警告：**如果在 %files 下列出一个目录名，RPM 会包括该目录下的所有文件。通常这不是您想要的，特别对于 /bin 这样的目录。

[**回页首**](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/index.html#ibm-pcon)

**避免简单实例中的问题**

这个最基本的 spec 文件有几个问题。最大的问题之一就是您最后在构建系统上实际安装了该产品。而这可能只是一个软件测试版本，您也许并不想在构建系统中安装它。

RPM 用一个名为 *构建根（build root）*的特性来处理这个问题。它的想法是设置您的 spec 文件，以将所有安装的文件复制到一个虚拟目录树（从构建根开始）；然后 RPM 从那里得到文件。

但是，这需要一些软件包的支持。在包括 indent 在内的很多 GNU 软件包中，在 make install 的时候定义 DESTDIR 将会在所有安装路径之前添加 DESTDIR 值。

请注意 **不**要使用 ./configure --prefix=$RPM\_BUILD\_ROOT 。这会在假设整个软件包文件的最终位置是构建根的情况下安装整个软件包。这对于 indent 可能没有关系，但任何需要在运行时找到其安装文件的程序都将失败，因为当 RPM 最终安装到用户系统后，这些文件就不再位于构建根之下 ― 那只是您构建系统上的一个临时目录。

请参阅更新的文件 [indent-2.spec](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/indent-2.spec)，如下所示。

**第二个 spec 文件：indent-2.spec**

Summary: GNU indent

Name: indent

Version: 2.2.6

Release: 2

Source0: %{name}-%{version}.tar.gz

License: GPL

Group: Development/Tools

BuildRoot: %{\_builddir}/%{name}-root

%description

The GNU indent program reformats C code to any of a variety of

formatting standards, or you can define your own.

%prep

%setup -q

%build

./configure

make

%install

rm -rf $RPM\_BUILD\_ROOT

make DESTDIR=$RPM\_BUILD\_ROOT install

%clean

rm -rf $RPM\_BUILD\_ROOT

%files

%defattr(-,root,root)

/usr/local/bin/indent

%doc /usr/local/info/indent.info

%doc %attr(0444,root,root) /usr/local/man/man1/indent.1

%doc COPYING AUTHORS README NEWS

**更改说明**

首先，我们增加了版本的发行号。无论何时，当您编辑 spec 文件时，都不要忘了这么做。

我们在头中添加了 BuildRoot，以便告诉 RPM 这是在构建期间临时安装文件的地方。对于临时文件，我们这里使用了两个 RPM 宏，而不是假设某个特定位置。在 Red Hat 7.1 上， %{\_builddir} 以类似于 /usr/src/redhat/BUILD 结束。

我们还需要告诉系统将 indent 安装在那里。RPM 帮助我们用构建根的值定义一个 shell 变量 RPM\_BUILD\_ROOT ，因此在 make install 时，我们只需将它作为 DESTDIR 值传入即可。

我们还在 %install 和 %clean 中添加了几行，以便在开始安装以前（为保险起见）和完成以后清除构建根。%clean 是一切都正常的情况下在 RPM 构建结束时运行的脚本，这样临时文件就不会一直保留。

最后，在 %files 中，请注意我们没有在此处的路径前包括 BuildRoot。我们使用了“真正”的路径；RPM 将在构建根下寻找这些文件，因为您已经包括了 BuildRoot 定义。

**这一次发生了什么**

如果仔细观察您会发现，在 RPM 进行安装部分以前，一切工作照旧。然后，文件将不直接安装到 /usr/local/bin，而是安装在（比如说）/usr/src/redhat/BUILD/indent-root/usr/local/bin 中。

如果您检查最终的二进制 RPM 文件（用 rpm -qlp indent-2.2.6-2.i386.rpm ），您会看到构建根已被 RPM 除去。如果您安装 RPM，这些文件最终将安装在正确的目录，如 /usr/local/bin/indent 中。

[**回页首**](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/index.html#ibm-pcon)

**在其它 Linux 分发版上使用 RPM**

如果您在使用不同的 Linux 分发版，RPM 可能会有不同的内置路径。例如，它几乎肯定不会在 /usr/src/redhat 查找源文件！要确定希望的 RPM 安装路径，请运行 rpm --showrc 并查看下列部分如何被定义：

**\_sourcedir**

RPM 在哪里查找源文件（tar 文件，等）

**\_srcrpmdir**

RPM 在哪里放入新的源 RPM 文件

**\_rpmdir**

RPM 将把新的二进制 RPM 文件放在哪里（在特定于体系结构的子目录中）

其中一些根据其它变量定义；例如，当您看到 %{\_topdir} ，查找 \_topdir 的定义，等等。

**[回页首](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/index.html" \l "ibm-pcon)**

**下一步是什么**

我希望这篇用 RPM 打包软件的介绍会对您有所帮助。有关相关的阅读材料，请参阅下面的 [参考资料](https://www.ibm.com/developerworks/cn/linux/management/package/rpm/part1/index.html#resources)。在本系列的后续文章中，我们将讨论这些主题：

* 构建 RPM 软件包而不必是 root 用户
* 在创建软件之前为软件打补丁
* 在安装和卸载时运行脚本
* 在安装或卸载 **其它**软件包时运行脚本