**011-04-02**

[RPM打包step by step(1)](http://www.worldhello.net/2011/04/02/2405.html)

最近学习rpm打包，参考ibm文档库里rpm打包的文章，结合自己的实践，总结如下，一来备忘，二来和大家交流。 和deb打包不同，rpm打包需要特定的目录及结构。查看rpm打包目录，以下为在CentOS5.5下的输出结果：

**$** rpm --showrc|grep \_topdir

-14: \_builddir %{\_topdir}/BUILD

-14: \_rpmdir %{\_topdir}/RPMS

-14: \_sourcedir %{\_topdir}/SOURCES

-14: \_specdir %{\_topdir}/SPECS

-14: \_srcrpmdir %{\_topdir}/SRPMS

-14: \_topdir %{\_usrsrc}/redhat

**$** rpm --showrc|grep \_usrsrc

-14: \_topdir %{\_usrsrc}/redhat

-14: \_usrsrc %{\_usr}/src

**$** rpm --showrc|grep \_usr

-14: \_defaultdocdir %{\_usr}/share/doc

-14: \_topdir %{\_usrsrc}/redhat

-14: \_usr /usr

-14: \_usrsrc %{\_usr}/src

经过层层寻找，最终发现打包目录在/usr/src/redhat目录下，看看目录结构：

$ tree /usr/src/redhat

/usr/src/redhat

|-- BUILD

|-- RPMS

| |-- athlon

| |-- geode

| |-- i386

| |-- i486

| |-- i586

| |-- i686

| `-- noarch

|-- SOURCES

|-- SPECS

`-- SRPMS

其中BUILD存放编译生成的临时文件，RPMS存放根据各种构架生成的rpm包，SOURCES存放源码包，SPECS存放spec文件，SRPMS存放生成的SRPM包。

**最简单例子**

下面以hello world为例，构建一个最小化打包过程。 首先需要写一个SPEC文件hello.spce：

Summary: hello world rpm package

Name: hello

Version: 0.1

Release: 1

Source: hello-0.1.tar.gz

License: GPL

Packager: amoblin

Group: Application

URL: http://www.ossxp.com

%description

This is a software for making your life more beautiful!

%prep

%setup -q

%build

gcc -o hello hello.c

%install

install -m 755 hello /usr/local/bin/hello

%files

/usr/local/bin/hello

放到上述SPECS目录下。 然后一个源程序hello.c：

#include <stdio.h>

int main()

{

printf("Hello, World!**\n**");

**return** 0;

}

存放在hello-0.1目录，然后打包放到SOURCES目录:

**$** tar zcvf hello-0.1.tar.gz hello-0.1

hello-0.1/

hello-0.1/hello.c

**$** sudo mv hello-0.1.tar.gz /usr/src/redhat/SOURCES

在SPECS目录下使用rpmbuild进行打包：

**$** cd /usr/src/redhat

**$** sudo rpmbuild -ba hello.spec

...

Wrote: /usr/src/redhat/SRPMS/hello-0.1-1.src.rpm

Wrote: /usr/src/redhat/RPMS/i386/hello-0.1-1.i386.rpm

这时会逐个运行hello.spec文件的内容，最终生成两个文件，一个包含源码的rpm包和一个二进制rpm包。 使用 -bs 选项只生成源码rpm包；使用 -bb 选项只生成rpm包。 查看rpm包信息和包内容：

**$** rpm -qpi ../RPMS/i386/hello-0.1-1.i386.rpm

**$** rpm -qpl ../RPMS/i386/hello-0.1-1.i386.rpm

第一个命令的输出是spec文件的序言部分的内容，第二个命令的输出是%files部分的文件列表。 现在有个问题，打包目录在/usr/src/redhat下，需要root权限才能操作，太不方便了，能不能在用户自定义目录下打包呢？

**自定义打包目录**

我们可以通过修改topdir宏的值来自定义打包路径：

**$** echo %\_topdir $HOME/rpmbuild > ~/.rpmmacros

这样再查看topdir的值会发现已变为用户主目录下rpm子目录了。这时修改文件就方便多了。 但在执行rpmbuild时仍会出问题：

**$** rpmbuild -ba hello.spec

...

install: 无法删除 “/usr/local/bin/hello”: 权限不够

error: Bad exit status from /var/tmp/rpm-tmp.65773 (%install)

...

这是因为rpmbuild在构建rpm包时会将程序安装一遍，然后再提取安装文件。由于需要复制二进制文件hello到系统目录/usr/local/bin/下，所以普通用户执行就报错了。 那么怎么办呢？这里需要使用虚拟根了。 修改spec文件，在Description段落前，URL字段之后增加一行：

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

修改install段落，将绝对安装路径改为使用构建根的方式：:

%install

mkdir -p $RPM\_BUILD\_ROOT/usr/local/bin

install -m 755 hello $RPM\_BUILD\_ROOT/usr/local/bin/hello

通过BuildRoot的值告诉rpmbuild，我们的构建根是builddir下的hello-root目录。其中以%{}括起来的是RPM宏，\_builddir代表~/rpmbuild/BUILD目录；name代表spec文件开头的Name字段值。 以下划线开头的builddir是系统RPM宏，我们可以通过rpm --showrc看到，可以在.rpmmacros中自定义。 RPM\_BUILD\_ROOT和前面的宏不同，这里没有{}括起来，是为了在以后安装生成的rpm时不至于也去寻找传说中的构建根。 如果喜欢的话，可以修改Source字段如下：:

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

好，继续回到构建根。现在执行rpmbuild，会在BUILD下创建hello-root目录作为虚拟根，hello安装在其中的usr/local/bin目录下。