

Construcción archivos src.rpm por Hoyt Duff

TUTORIAL: [Construyendo desde archivos src.rpm - Modificando "Freetype"](http://fedoranews.org/hoyt/rpm/)
(<http://fedoranews.org/hoyt/rpm/>)
Por Hoyt Duff el 26-01-2004

Desde que leíste HOWTO: How to create rpmbuild directory tenías una construcción instalada basada, toda ella, en RPM, ahora vamos a ponerla en uso. (Nota: “fedora-buildrpm-tree” no crea la arquitectura específica de sub-directorios en /BUILD, pero no os preocupeis, porque esto no supone ningún problema).

La referencia canónica para RPM es el libro “Maximum RPM” (tristemente anticuado, la última actualización fue en el 2000), pero todavía es una buena referencia y se hace necesaria su lectura para construcciones serias de RPM.

Allí puedes encontrar tres lugares (cuatro actualmente, si usas la opción de especificar cualquier localización aleatoria) que la configuración de la construcción RPM guarda. Las macros de la arquitectura específica son guardados en: /usr/lib/rpm//macros. Encontrarás también los archivos macros en el directorio /redhat. Aquel parece la plantilla principal, llena de variables que los otros archivos del config fijaron. Es bastante recomendable que no modifiques ninguno de estos archivos hasta que seas capaz hacer cambios con una entrada a ~/.rpmmacros.

Si quieres hacer algún cambio en el “system-level”, estarán en /etc/rpm/macros.*; los archivos son 'macros.prelink', 'macros.solve', 'macros.specspo', y 'macros.up2date' más un archivo adicional denominado 'platform'. No estoy completamente seguro de la fuente que explica el funcionamiento de estos archivos en /etc/rpm.

Los cambios específicos que hace el usuario se guardan en ~/.rpmmacros. Puedes, por ejemplo, añadir "%optflags-funroll-loops" al archivo ~/.rpmmacros y luego "rpmbuild --showrc | grep unroll" para ver el “optflag” formar parte del ambiente (hacer esto antes y después para ver el cambio; mirar también usando --showrc. Fíjate en que puedes tener varios archivos .rpmmacro y especifica cual quieres usar con el directorio --rcfile).

Los archivos están comentados (sin embargo los archivos .rpmmacros no lo están), por tanto puedes ponerlos en un procesador de texto y ver que es lo que hacen. Puedes también referirte a ellos en “Appendix B in Maximum RPM” en: <http://ftp.redhat.com/pub/redhat/linux/9/en/doc/RH-DOCS/maximum-rpm-1.0/html/ch-rpmrc-file.html>.

Las páginas del hombre para “rpm, rpmbuild, y gcc” serán prácticas.

Puedes ver los valores prefijados actuales con:

```
$ rpmbuild --showrc
```

El formato de las entradas es el siguiente:

```
name value
```

Cuando arranques “fedora-buildtree”, te creará un archivo `~.rpmmacros` con los siguientes contenidos:

```
%_topdir %(echo $HOME)/rpmbuild
```

Puede que quieras añadir:

```
%_tmppath %(echo $HOME)/tmp
```

Para dirigirlo hacia tu propio directorio `/tmp`, y

```
%optflags -march=athlon  
%_vendor fedora
```

Y cualquier otra variable que creas apropiada. Con la opción `--rcfile`, puedes experimentar con diferentes `.rpmmacros`.

Construyendo un archivo RPM desde un archivo `*.src.rpm`

Usaremos un ejemplo muy típico para recompilar el archivo `src.rpm` para “Freetype” de modo que el bytecode normalmente deshabilitado es habilitado, proveyéndonos una fuente más agradable para la vista.

Paso 1

Instala el `freetype-2.1.4-5.src.rpm`. Lo pondrá en tu directorio `~/rpmbuild/SOURCES` y `~/rpmbuild/SPECS`. Lo pondrá en tu fuente `rpms` como `non-root`.

Paso 2

Abre el archivo `.spec` en el subdirectorio `/SPECS` en cualquier procesador de texto. Las tres primeras líneas se parecerán a esto:

```
# Disables freetype 1 and 2 bytecode interpreter. Setting to 0 enables # the  
bytecode interpreter in both freetype 1 and 2. %define  
without_bytecode_interpreter 1
```

Cambia el valor de ‘1’ a ‘0’ para habilitar el intérprete bytecode, salva y cierra el archivo. Es una buena idea que anotes en el campo de descripción lo que has cambiado.

Paso 3

Construye el nuevo RPM con:

```
$ rpmbuild -bb ~/rpmbuild/SPECS/freetype.spec
```

y encontrarás nuevos archivos binarios RPM en el subdirectorio /RPM. Nota: Si quieres eliminar la opción de la arquitectura, por ejemplo, puedes poner esto en el comando:

```
$ rpmbuild -bb --target=athlon ~/rpmbuild/SPECS/freetype.spec
```

Paso 4

Instala el nuevo archivo RPM con:

```
# rpm -Uvh --replacepks ~/rpmbuild//freetype-2.1.4-5*rpm
```

El '--replacepks' te permite “actualizar” la misma versión del paquete.

Paso 5

Actualmente, conocemos la variable llamada 'without_bytecode_interpreter', podemos reconstruirla con el archivo src.rpm pasando el valor cambiado a la línea del comando de este modo:

```
$ rpmbuild -bb --with without_bytecode_interpreter=0  
~/rpmbuild/SPECS/freetype.spec
```

Como puedes ver RPM es poderosa y flexible. Que lo disfrutes!

LINKS

RPM.Org - <http://www.rpm.org/>

Maximum RPM download- <http://www.redhat.com/docs/books/max-rpm/max-rpm.pdf>

Maximum RPM - <http://ftp.redhat.com/pub/redhat/linux/9/en/doc/RH-DOCS/maximum-rpm-1.0/html/>

Fedora RPMDev Tools - <http://www.fedora.us/wiki/FedoraRPMDevTools>

Fedora RPMDev Tools download -
<http://download.fedora.us/fedora/fedora/1/i386/RPMS.stable/fedora-rpmdevtools-0.1.6-0.fdr.1.1.noarch.rpm>

RPM related documentation - file:/usr/share/doc/rpm-devel-4.2.1/apidocs/html/pages.html Write to me at hoyt@maximumhoyt.com.

Copyright © 2004 Hoyt Duff. Permission granted for use on FedoraNEWS.ORG. All rights

reserved.

Traducido por Vassago para Fent Linux

Liberada bajo licencia



<http://creativecommons.org/licenses/by-nc-sa/2.5/>