RPM パッケージの操作と構築

2013年3月6日

OSS基盤技術センター
OSS技術第二課

パッケージ管理の種類

パッケージは対象とするソフトの必要なライブラリ、設定ファイルをまとめて配布する形式。

Linux ではディストリビューションによって各々独自のパッケージ管理を行っている。現状では主に下記の種類がある。

- Slackware: tar ball 形式

  slackware, SUSE, Plamo , …

- Debian系 : deb 形式

  Debian, Ubuntu, Knopix, …

- RedHat系 : rpm形式

  RedHat, Vine, TurboLinux, SUSE, …

以降では CentOS6.2 環境でのパケージ管理について説明します。

パッケージの種別

パッケージには2種類ある

- バイナリパッケージ

  実行ファイル、ライブラリを格納するインストール用パッケージ。アーキテクチャ毎に存在する。

  package-name-{version}-{release}.{arch}.rpm

- ソースパッケージ

  パッケージビルドの為に使用するソースファイル、specファイルを格納するパッケージ。

  package-name-{version}-{release}.src.rpm

パッケージ操作

- RedHat 系のディストリビューションで使用するパッケージ操作コマンド

  rpm、yum、rpmbuild

- rpm (RPM Package Manager)

  単純なパッケージ操作コマンド

  - パッケージは自分で用意する必要がある。
  - rpm のパッケージインストールではパッケージファイルを直接指定する。
  - 依存関係は自分で解決する必要がある。

- yum(Yellowdog Updater Modified)

  rpm より使い勝手が良い

  - 依存関係を解決してくれる。
  - ネットワークリポジトリ上のパッケージを参照できる。
  - パッケージを検索できる。
  - パッケージグループが扱える。
- rpmbuild

  パッケージの構築に使用

rpmでの操作

インストール関連の操作

機能	コマンド
新規インストール	rpm -i package-file-name
既インストール済みのパッケージのアップグレード（但し、未インストールのパッケージは新規としてインストールされる）	rpm -U package-file-name
同バージョンのパッケージの再インストール	rpm -U --replacepkgs package-file-name
パッケージのダウングレード	rpm -U --oldpackage package-file-name
既インストール済みのパッケージのみアップグレード（未インストールのパッケージはインストールされ無い）	rpm -F package-file-name
パッケージのアンインストール	rpm -e package-name

インストール関連の操作(続き)

機能	コマンド
依存関係を無視してインストール	rpm -i --nodeps package-file-name
インストール先の変更(再配置可能なパッケージの場合)	rpm -i --prefix new-path package-file-name rpm -i --reloacte old-path=new-path package-file-name
再配置不可能なパッケージのインストール先の変更	rpm -i --relocate old-path=new-path --badreloc package-file-name
指定したディレクトリを root としてインストール	rpm -i --root root-path package-file-name
インストール動作の確認	rpm -ivv --test package-file-name

インストール操作の注意点

- インストールする際には対象とするパッケージの依存関係を満たす必要がある。

  そのため必要となるパッケージは予めインストールしておくか対象パッケージと一緒にインストールする必要がある。
- 再配置可能なパッケージとは spec ファイル中で対象となるディレクトリが ‘Prefix:’ ディレクティブで指定されているパッケージの事。

➢ 再配置不可能なパッケージのインストール先の変更の例

```
$ sudo rpm -ivh --relocate /sbin=/usr/local/sbin device-mapper-multipath-0.4.9-46.el6.x86_64.rpm device-
mapper-multipath-libs-0.4.9-46.el6.x86_64.rpm
Preparing...                ########################################### [100%]
      path /sbin in package device-mapper-multipath-libs-0.4.9-46.el6.x86_64 is not relocatable
      path /sbin in package device-mapper-multipath-0.4.9-46.el6.x86_64 is not relocatable
$ rpm -qa | grep device-mapper-multipath
$ sudo rpm -ivh --relocate /sbin=/usr/local/sbin --badreloc device-mapper-multipath-0.4.9-
46.el6.x86_64.rpm device-mapper-multipath-libs-0.4.9-46.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:device-mapper-multipath########################################### [ 50%]
   2:device-mapper-multipath########################################### [100%]
$ rpm -qa | grep device-mapper-multipath
device-mapper-multipath-libs-0.4.9-46.el6.x86_64
device-mapper-multipath-0.4.9-46.el6.x86_64
$ ls /usr/local/sbin
cciss_id  mpathconf  multipath  multipathd
$ ls /sbin/cciss_id /sbin/mpathconf /sbin/multipath /sbin/multipathd
ls: cannot access /sbin/cciss_id: No such file or directory
ls: cannot access /sbin/mpathconf: No such file or directory
ls: cannot access /sbin/multipath: No such file or directory
ls: cannot access /sbin/multipathd: No such file or directory
$
```

- パッケージが複数バージョンインストールされている時の削除の例

```
$ sudo rpm --root /tmp/rpmtest -ivh /data1/packages/rpms/co62/kpartx-0.4.9-46.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:kpartx                 ########################################### [100%]
$ sudo rpm --root /tmp/rpmtest -ivh --prefix /usr/local/sbin kpartx-0.4.9-46.el6.vaj1.x86_64.rpm
Preparing...                ########################################### [100%]
   1:kpartx                 ########################################### [100%]
$ sudo rpm -qa --root /tmp/rpmtest | grep kpartx
 kpartx-0.4.9-46.el6.vaj1.x86_64
kpartx-0.4.9-46.el6.x86_64
$ sudo rpm --root /tmp/rpmtest -ev kpartx
error: "kpartx" specifies multiple packages:
  kpartx-0.4.9-46.el6.x86_64
  kpartx-0.4.9-46.el6.vaj1.x86_64
$ sudo rpm --root /tmp/rpmtest -ev kpartx-0.4.9-46.el6.vaj1.x86_64
$ sudo rpm -qa --root /tmp/rpmtest | grep kpartx
kpartx-0.4.9-46.el6.x86_64
$ sudo rpm --root /tmp/rpmtest -ivh --prefix /usr/local/sbin kpartx-0.4.9-46.el6.vaj1.x86_64.rpm
Preparing...                ########################################### [100%]
   1:kpartx                 ########################################### [100%]
$ sudo rpm --root /tmp/rpmtest -ev --allmatches kpartx
$ sudo rpm -qa --root /tmp/rpmtest | grep kpartx
$
```

パッケージ情報の参照

機能	コマンド
パッケージの詳細情報の表示	rpm -qi package-name
依存情報の表示	rpm -qR package-name
指定パッケージに依存している情報の表示	rpm -q --whatrequires package-name
パッケージの実行スクリプト	rpm -q --scripts package-name
パッケージに格納されている設定ファイルの表示	rpm -qc package-name
パッケージに格納されている文書ファイルの表示	rpm -qd package-name
パッケージに格納されている全ファイルの表示	rpm -ql package-name
パッケージの一覧	rpm -qa

パッケージ情報の参照(続き)

機能	コマンド
パッケージファイルを指定したパッケージ情報の参照。	rpm -qp {options} package-file-name
パッケージのグループ情報の表示	rpm -qg group-name
所属パッケージの表示	rpm -qf file-path
パッケージの展開	rpm2cpio package-file-name | cpio -id

グループ名は “/usr/doc/share/rpm-{version}/ GROUPS” ファイルを参照する事で確認できる。

ここで言うグループは後述の yum で扱うパッケージグループとは異なり、specファイル内で定義されたパッケージの種別となる。

- -qip の例

```
$ rpm -qip device-mapper-multipath-0.4.9-46.el6.x86_64.rpm
Name        : device-mapper-multipath      Relocations: (not relocatable)
Version     : 0.4.9                         Vendor: CentOS
Release     : 46.el6                       Build Date: Thu 08 Dec 2011 09:05:42 AM JST
Install Date: (not installed)              Build Host: c6b18n2.bsys.dev.centos.org
Group       : System Environment/Base      Source RPM: device-mapper-multipath-0.4.9-46.el6.src.rpm
Size        : 184931                        License: GPL+
Signature   : RSA/SHA1, Fri 09 Dec 2011 04:54:13 AM JST, Key ID 0946fca2c105b9de
Packager    : CentOS BuildSystem <http://bugs.centos.org>
URL         : http://christophe.varoqui.free.fr/
Summary     : Tools to manage multipath devices using device-mapper
Description :
device-mapper-multipath provides tools to manage multipath devices by
instructing the device-mapper multipath kernel module what to do.
The tools are :
* multipath - Scan the system for multipath devices and assemble them.
•multipathd - Detects when paths fail and execs multipath to update things.
$
```

- -qR の例

```
$ rpm -qR device-mapper-multipath-0.4.9-46.el6.x86_64
/bin/bash
/bin/sh
/bin/sh
/bin/sh
/bin/sh
chkconfig
```

```
chkconfig
config(device-mapper-multipath) = 0.4.9-46.el6
device-mapper >= 1.02.38
device-mapper-multipath-libs = 0.4.9-46.el6
initscripts
initscripts
kpartx = 0.4.9-46.el6
libc.so.6()(64bit)
libc.so.6(GLIBC_2.2.5)(64bit)
libc.so.6(GLIBC_2.3)(64bit)
libc.so.6(GLIBC_2.7)(64bit)
libdevmapper.so.1.02()(64bit)
libdevmapper.so.1.02(Base)(64bit)
libdl.so.2()(64bit)
libmultipath.so()(64bit)
libncurses.so.5()(64bit)
libpthread.so.0()(64bit)
libpthread.so.0(GLIBC_2.2.5)(64bit)
libpthread.so.0(GLIBC_2.3.2)(64bit)
libpthread.so.0(GLIBC_2.3.3)(64bit)
libreadline.so.6()(64bit)
rpmlib(CompressedFileNames) <= 3.0.4-1
rpmlib(FileDigests) <= 4.6.0-1
rpmlib(PayloadFilesHavePrefix) <= 4.0-1
rtld(GNU_HASH)
rpmlib(PayloadIsXz) <= 5.2-1
$
```

- --whatrequires の例

```
$ rpm -q --whatrequires device-mapper-multipath-libs
device-mapper-multipath-0.4.9-46.el6.x86_64
$
```

- -q --scripts の例

```
$ rpm -q --scripts device-mapper-multipath-0.4.9-46.el6.x86_64
postinstall scriptlet (using /bin/sh):
/sbin/chkconfig --add multipathd
if [ "$1" -gt "1" -a ! -e /etc/multipath/bindings -a ¥
   -f /var/lib/multipath/bindings ]; then
      mv /var/lib/multipath/bindings /etc/multipath/bindings
      ln -s /etc/multipath/bindings /var/lib/multipath/bindings
fi
preuninstall scriptlet (using /bin/sh):
if [ "$1" = 0 ]; then
      /sbin/service multipathd stop /dev/null 2>&1
      /sbin/chkconfig --del multipathd
fi
postuninstall scriptlet (using /bin/sh):
if [ "$1" -ge "1" ]; then
      /sbin/service multipathd condrestart >/dev/null 2>&1 || :
fi
$
```

- -ql, -qf の例

```
$ rpm -ql device-mapper-multipath-0.4.9-46.el6.x86_64
/etc/multipath
/etc/rc.d/init.d/multipathd
/lib/udev/rules.d/40-multipath.rules
/sbin/cciss_id
/sbin/mpathconf
/sbin/multipath
/sbin/multipathd
/usr/share/doc/device-mapper-multipath-0.4.9
/usr/share/doc/device-mapper-multipath-0.4.9/AUTHOR
/usr/share/doc/device-mapper-multipath-0.4.9/COPYING
/usr/share/doc/device-mapper-multipath-0.4.9/FAQ
/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf
/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.annotated
/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.defaults
/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf.synthetic
/usr/share/man/man5/multipath.conf.5.gz
/usr/share/man/man8/mpathconf.8.gz
/usr/share/man/man8/multipath.8.gz
/usr/share/man/man8/multipathd.8.gz
$ rpm -qf /sbin/multipath
device-mapper-multipath-0.4.9-46.el6.x86_64
$
```

- rpm2cpio の例

```
$ rpm2cpio ../device-mapper-multipath-0.4.9-46.el6.x86_64.rpm | cpio -id
368 blocks
$ ls -R
.:
etc  lib  sbin  usr

./etc:
multipath  rc.d

./etc/multipath:

./etc/rc.d:
init.d

./etc/rc.d/init.d:
multipathd

./lib:
udev

./lib/udev:
rules.d

./lib/udev/rules.d:
40-multipath.rules

./sbin:
cciss_id  mpathconf  multipath  multipathd
```

- rpm2cpio の例(続き)

```
./usr:
share

./usr/share:
doc  man

./usr/share/doc:
device-mapper-multipath-0.4.9

./usr/share/doc/device-mapper-multipath-0.4.9:
AUTHOR   FAQ          multipath.conf.annotated  multipath.conf.synthetic
COPYING  multipath.conf  multipath.conf.defaults

./usr/share/man:
man5  man8

./usr/share/man/man5:
multipath.conf.5.gz

./usr/share/man/man8:
mpathconf.8.gz  multipath.8.gz  multipathd.8.gz
$
```

- パッケージのグループ情報

```
$ cat /usr/share/doc/rpm-4.8.0/GROUPS
Amusements/Games
Amusements/Graphics
Applications/Archiving
Applications/Communications
Applications/Databases
Applications/Editors
Applications/Emulators
Applications/Engineering
Applications/File
Applications/Internet
Applications/Multimedia
Applications/Productivity
Applications/Publishing
…
$ rpm -qg Applications/Archiving
cpio-2.10-9.el6.x86_64
tar-1.23-3.el6.x86_64
wodim-1.1.9-11.el6.x86_64
k3b-common-1.0.5-13.el6.noarch
k3b-1.0.5-13.el6.x86_64
zip-3.0-1.el6.x86_64
unzip-6.0-1.el6.x86_64
$
```

yumでの操作

yumの設定

- 設定ファイル

  /etc/yum.conf

  /etc/yum.repos.d/*.repo

  /etc/yum/pluginconf.d/*.conf

  /etc/yum/protected.d/*.conf

- /etc/yum.conf
  - yum全般の設定
  - リポジトリの設定

    /etc/yum.repos.d/*.repo に設定がある場合はマージされる

- /etc/yum.conf の例

```
[main]
cachedir=/var/cache/yum/$basearch/$releasever
keepcache=1
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit=5
bugtracker_url=http://bugs.centos.org/set_project.php?project_id=16&ref=http://bugs.centos.org/bug_report_page.php?category=yum
distroverpkg=centos-release

#  This is the default, if you make this bigger yum won't see if the metadata
# is newer on the remote and so you'll "gain" the bandwidth of not having to
# download the new metadata and "pay" for it by yum not having correct
# information.
#  It is esp. important, to have correct metadata, for distributions like
# Fedora which don't keep old packages around. If you don't like this checking
# interupting your command line usage, it's much better to have something
# manually check the metadata once an hour (yum-updatesd will do this).
# metadata_expire=90m

# PUT YOUR REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d
```

yumの設定(続き)

- /etc/yum.repos.d/*.repo

  後述

- /etc/yum/pluginconf.d/*.conf
  - yum には多数のプラグインがあり機能追加出来る。
  - インストールした際にプラグインの設定情報がここに置かれる

    yum-plugin-{plugin-name}-{version等} と言うプラグインの設定ファイルはplugin-name.confと言うファイル名で作成。

- /etc/yum/protected.d/*.conf

  対象のパッケージとその依存パッケージを削除する事からプロテクトする。

  任意の名前で作成し(例えば kakuma.conf)プロテクトしてほしいパッケージ名を書く。

➢ プラグイン一覧

```
$ yum search yum-plugin
Loaded plugins: downloadonly, fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
 * base: mirror.fairway.ne.jp
 * extras: mirror.fairway.ne.jp
 * updates: mirror.fairway.ne.jp
================================== N/S Matched: yum-plugin
==================================
PackageKit-yum-plugin.x86_64 : Tell PackageKit to check for updates when yum exits
anaconda-yum-plugins.noarch : Installation-related yum plugins
kabi-yum-plugins.noarch : The CentOS Linux kernel ABI yum plugin
yum-plugin-aliases.noarch : Yum plugin to enable aliases filters
yum-plugin-auto-update-debug-info.noarch : Yum plugin to enable automatic updates to installed
                                         : debuginfo packages
yum-plugin-changelog.noarch : Yum plugin for viewing package changelogs before/after updating
yum-plugin-downloadonly.noarch : Yum plugin to add downloadonly command option
yum-plugin-fastestmirror.noarch : Yum plugin which chooses fastest repository from a mirrorlist
yum-plugin-filter-data.noarch : Yum plugin to list filter based on package data
yum-plugin-fs-snapshot.noarch : Yum plugin to automatically snapshot your filesystems during updates
yum-plugin-keys.noarch : Yum plugin to deal with signing keys
yum-plugin-list-data.noarch : Yum plugin to list aggregate package data
yum-plugin-local.noarch : Yum plugin to automatically manage a local repo. of downloaded packages
yum-plugin-merge-conf.noarch : Yum plugin to merge configuration changes when installing packages
yum-plugin-post-transaction-actions.noarch : Yum plugin to run arbitrary commands when certain pkgs
                                           : are acted on
yum-plugin-priorities.noarch : plugin to give priorities to packages from different repos
```

➢ プラグイン一覧（続き）

```
yum-plugin-protect-packages.noarch : Yum plugin to prevents Yum from removing itself and other
                                   : protected packages
yum-plugin-protectbase.noarch : Yum plugin to protect packages from certain repositories.
yum-plugin-ps.noarch : Yum plugin to look at processes, with respect to packages
yum-plugin-remove-with-leaves.noarch : Yum plugin to remove dependencies which are no longer used
                                     : because of a removal
yum-plugin-rpm-warm-cache.noarch : Yum plugin to access the rpmdb files early to warm up access to
                                 : the db
yum-plugin-security.noarch : Yum plugin to enable security filters
yum-plugin-show-leaves.noarch : Yum plugin which shows newly installed leaf packages
yum-plugin-tmprepo.noarch : Yum plugin to add temporary repositories
yum-plugin-tsflags.noarch : Yum plugin to add tsflags by a commandline option
yum-plugin-upgrade-helper.noarch : Yum plugin to help upgrades to the next distribution version
yum-plugin-verify.noarch : Yum plugin to add verify command, and options
yum-plugin-versionlock.noarch : Yum plugin to lock specified packages from being updated

  Name and summary matches only, use "search all" for everything.
$
```

➢ プラグインのhelp

```
$ yum help
Loaded plugins: downloadonly, fastestmirror, refresh-packagekit, security
Usage: yum [options] COMMAND

List of Commands:
...
  Plugin Options:
    --downloadonly      don't update, just download
    --downloaddir=DLDIR
                        specifies an alternate directory to store packages
    --security          Include security relevant packages
    --bugfixes          Include bugfix relevant packages
    --cve=CVE           Include packages needed to fix the given CVE
    --bz=BZ             Include packages needed to fix the given BZ
    --sec-severity=SEVERITY
                        Include security relevant packages, of this severity
    --advisory=ADVISORY
                        Include packages needed to fix the given advisory
$
```

主なプラグイン

- yum-plugin-fastestmirror
  コマンド起動時に速いリポジトリを選択
- yum-plugin-downloadony
  yum のキャッシュディレクトリにダウンロード
  (/var/cache/yum/x86_64/6/base/packages/ 等)
- yum-plugin-priorities
  リポジトリの優先度を設定できる。
  .repo ファイルに priority=x の形で指定。

➢ ダウンロードプラグインの例

```
$ sudo yum install --downloadonly device-mapper-multipath
Loaded plugins: downloadonly, fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
 * base: ftp.nara.wide.ad.jp
 * extras: ftp.nara.wide.ad.jp
 * updates: ftp.nara.wide.ad.jp
base                                                     | 3.7 kB     00:00
extras                                                   | 3.5 kB     00:00
updates                                                    | 3.5 kB     00:00
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package device-mapper-multipath.x86_64 0:0.4.9-46.el6 will be updated
---> Package device-mapper-multipath.x86_64 0:0.4.9-56.el6_3.1 will be an update
--> Processing Dependency: kpartx = 0.4.9-56.el6_3.1 for package: device-mapper-multipath-0.4.9-
56.el6_3.1.x86_64
--> Processing Dependency: device-mapper-multipath-libs = 0.4.9-56.el6_3.1 for package: device-mapper-
multipath-0.4.9-56.el6_3.1.x86_64
--> Running transaction check
---> Package device-mapper-multipath-libs.x86_64 0:0.4.9-46.el6 will be updated
---> Package device-mapper-multipath-libs.x86_64 0:0.4.9-56.el6_3.1 will be an update
---> Package kpartx.x86_64 0:0.4.9-46.el6 will be updated
---> Package kpartx.x86_64 0:0.4.9-56.el6_3.1 will be an update
--> Finished Dependency Resolution
```

➢ ダウンロードプラグインの例(続き)

```
Dependencies Resolved

================================================================================
 Package                Arch       Version              Repository    Size
================================================================================
Updating:
 device-mapper-multipath          x86_64      0.4.9-56.el6_3.1        updates       96 k
Updating for dependencies:
 device-mapper-multipath-libs     x86_64      0.4.9-56.el6_3.1        updates      158 k
 kpartx                           x86_64      0.4.9-56.el6_3.1        updates       51 k

Transaction Summary
================================================================================
Upgrade       3 Package(s)

Total download size: 305 k
Is this ok [y/N]: y
Downloading Packages:
(1/3): device-mapper-multipath-0.4.9-56.el6_3.1.x86_64.rpm            |  96 kB     00:00
(2/3): device-mapper-multipath-libs-0.4.9-56.el6_3.1.x86_64.rpm        | 158 kB     00:00
(3/3): kpartx-0.4.9-56.el6_3.1.x86_64.rpm                              |  51 kB     00:00
--------------------------------------------------------------------------------
Total                                       1.6 MB/s | 305 kB     00:00


exiting because --downloadonly specified
$ ls /var/cache/yum/x86_64/6/updates/packages/
device-mapper-multipath-0.4.9-56.el6_3.1.x86_64.rpm       kpartx-0.4.9-56.el6_3.1.x86_64.rpm
device-mapper-multipath-libs-0.4.9-56.el6_3.1.x86_64.rpm
$
```

yumのリポジトリ

- リポジトリはバイナリパッケージ、ソースパッケージを管理するパッケージプール。
- ネットワーク経由でリポジトリにアクセスできる。
- パッケージグループ情報も設定できる。
- リポジトリ自体はコマンドで簡単に作成できる。
- リポジトリの指定は /etc/yum.repos.d/*.repo又は /etc/yum.conf で行う。

- /etc/yum.repos.d/*.repo
  - .repo ファイル毎にリポジトリ個々の設定を持つ。

    /etc/yum.conf　に設定がある場合はマージされる。
  - ここで指定されたリポジトリの中で最新のバージョンのパッケージがインストールされる。
  - サードパーティ製のパッケージを利用する場合、独自リポジトリを利用する場合もここで定義しておけば良い。
  - オプション例

    exclude=kernel* kpartx* …

    を指定した場合指定パッケージをインストール、アップデートの対象としない。

    これにより特定のリポジトリだけでアップデートを許す(もしくは許さない)様な指定もできる。

- /etc/yum.repos.d/CentOS-Base.repo

```
# CentOS-Base.repo
#
# The mirror system uses the connecting IP address of the client and the
# update status of each mirror to pick mirrors that are updated to and
# geographically close to the client.  You should use this for CentOS updates
# unless you are manually picking other mirrors.
#
# If the mirrorlist= does not work for you, as a fall back you can try the
# remarked out baseurl= line instead.
#
#

[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

#released updates
[updates]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

- /etc/yum.repos.d/CentOS-Base.repo(続き)

```
#additional packages that may be useful
[extras]
name=CentOS-$releasever - Extras
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=extras
#baseurl=http://mirror.centos.org/centos/$releasever/extras/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

#additional packages that extend functionality of existing packages
[centosplus]
name=CentOS-$releasever - Plus
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=centosplus
#baseurl=http://mirror.centos.org/centos/$releasever/centosplus/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

#contrib - packages by Centos Users
[contrib]
name=CentOS-$releasever - Contrib
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=contrib
#baseurl=http://mirror.centos.org/centos/$releasever/contrib/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

- /etc/yum.repos.d/CentOS-Media.repo

```
# CentOS-Media.repo
#
# This repo is used to mount the default locations for a CDROM / DVD on
#  CentOS-6.  You can use this repo and yum to install items directly off the
#  DVD ISO that we release.
#
# To use this repo, put in your DVD and use it with the other repos too:
#  yum --enablerepo=c6-media [command]
#
# or for ONLY the media repo, do this:
#
#  yum --disablerepo=¥* --enablerepo=c6-media [command]

[c6-media]
name=CentOS-$releasever - Media
baseurl=file:///media/CentOS/
        file:///media/cdrom/
        file:///media/cdrecorder/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

●リポジトリの作成

カスタムリポジトリは下記の手順で作成できる

■ リポジトリに含めるパッケージを何れかのツリーに集める
(ここでは/var/repo とする)

■ 必要であればパケージグループファイルを置く
(ここでは/var/repo/repomd.xml とする)

◆ファイルはxml形式

◆パケージグループファイルは一から作成しても良いが面倒なら、インストールCD内の repodata/repomd.xml をコピーして修正すれば良い。

■ createrepo コマンドで作成(createrepo-0.9.8-4.el6.noarch)。

```
$ createrepo -g /var/repo/repomd.xml /var/repo
```

■ パッケージツリーをネットワークに公開。

●パッケージグループとは

- ある機能に関連する複数のパッケージを依存関係ではなく、まとめて扱う様に定義されたパッケージ群。
- パッケージグループの情報はリポジトリ内に xml 形式のファイルとして置かれている。
- yum-groups-manager　を使用するとある程度の定義を追加できる。全項目を網羅していない様なので、テンプレートの生成的に使用できる。

➢パッケージグループのxmlファイル例

```
<!DOCTYPE comps PUBLIC "-//CentOS//DTD Comps info//EN" "comps.dtd">
<comps>
 <group>
  <id>group-id</id>
  <name>group-name</name>
  <name xml:lang='xx'>group-name-by-lang</name>
  <description>groupdescriptions.</description>
  <description xml:lang='xx'>description-by-lang</description>
  ...
  <default>true</default>
  <uservisible>true</uservisible>
  <packagelist>
    <packagereq type="mandatory">package-name</packagereq>
    <packagereq type="default">package-name</packagereq>
    <packagereq type="optional">package-name</packagereq>
    ...
   </packagelist>
 </group>
 <category>
  <id>category-id</id>
  <name>category-name</name>
  <name xml:lang='xx'>category-name-by-lang</name>
  ...
  <description>category-descriptions.</description>
  <description xml:lang='ja'>descriptions-by-lang</description>
  <grouplist>
  ...
   <groupid>group-id</groupid>
   ...
  </grouplist>
 </category>

</comps>
```

➢ パッケージグループのxml ファイル生成の例

```
$ yum-groups-manager -n "System Utility Tools" --id sys-util-tools --save comps.xml --description 'system
utility tools' --mandatory strace sysstat strace --optional bridge-utils
Loaded plugins: fastestmirror, priorities, refresh-packagekit
$ cat comps.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE comps PUBLIC "-//Red Hat, Inc.//DTD Comps info//EN" "comps.dtd">
<comps>

 <group>
  <id>sys-util-tools</id>
  <default>false</default>
  <uservisible>true</uservisible>
  <display_order>1024</display_order>
  <name>System Utility Tools</name>
  <description>system utility tools</description>
   <packagelist>
    <packagereq type="mandatory">bridge-utils</packagereq>
    <packagereq type="mandatory">strace</packagereq>
    <packagereq type="mandatory">sysstat</packagereq>
   </packagelist>
 </group>
</comps>
$
```

➢ パッケージグループのxml ファイルのマージの例

```
$ yum-groups-manager -n "System Debug Tools" --id sys-debug-tools --merge comps.xml --description 'system debug tools' --mandatory
valgrind --optional crash
Loaded plugins: fastestmirror, priorities, refresh-packagekit
[test@centos62_1 groups]$ cat comps.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE comps PUBLIC "-//Red Hat, Inc.//DTD Comps info//EN" "comps.dtd">
<comps>

 <group>
  <id>sys-debug-tools</id>
  <default>false</default>
  <uservisible>true</uservisible>
  <display_order>1024</display_order>
  <name>System Debug Tools</name>
  <description>system debug tools</description>
   <packagelist>
    <packagereq type="mandatory">crash</packagereq>
    <packagereq type="mandatory">valgrind</packagereq>
   </packagelist>
 </group>
 <group>
  <id>sys-util-tools</id>
  <default>false</default>
  <uservisible>true</uservisible>
  <display_order>1024</display_order>
  <name>System Utility Tools</name>
  <description>system utility tools</description>
   <packagelist>
    <packagereq type="mandatory">bridge-utils</packagereq>
    <packagereq type="mandatory">strace</packagereq>
    <packagereq type="mandatory">sysstat</packagereq>
   </packagelist>
 </group>
</comps>
$
```

yumのオプション

オプション	機能
config ファイルの指定	-c, --config=config-file
リポジトリの有効化	--enablerepo=repo-id
リポジトリの無効化	--disablerepo=repo-id
GPGシグニチャのチェックを行わない	--nogpgcheck
指定したディレクトリを root としてインストール	--installroot= root-path
指定パッケージのアップデートを行わない	-x, --exclude=package-name

インストール関連の操作

機能	コマンド
新規インストール	yum install package-name
同バージョンのパッケージの再インストール	yum reinstall package-name
パッケージのダウングレード	yum downgrade package-name
既インストール済みのパッケージのアップデート（但し、未インストールのパッケージは新規としてインストールされる）	yum install package-name
既インストール済みのパッケージのみアップデート（未インストールのパッケージはインストールされ無い）	yum update [package-name]
パッケージのアンインストール	yum erase package-name yum remove package-name

- rpmコマンドの様にパッケージファイルを直接指定しても良い。この場合依存パッケージがあればリポジトリからインストールしてくれる。

  明示的なローカルインストールサブコマンド localinstall、localupdateを指定しても同様。

➢ パッケージインストールの例

```
$ sudo yum update device-mapper-multipath
Loaded plugins: downloadonly, fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
 * base: ftp.iij.ad.jp
 * extras: ftp.iij.ad.jp
 * updates: ftp.iij.ad.jp
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package device-mapper-multipath.x86_64 0:0.4.9-46.el6 will be updated
---> Package device-mapper-multipath.x86_64 0:0.4.9-56.el6_3.1 will be an update
--> Processing Dependency: kpartx = 0.4.9-56.el6_3.1 for package: device-mapper-multipath-0.4.9-56.el6_3.1.x86_64
--> Processing Dependency: device-mapper-multipath-libs = 0.4.9-56.el6_3.1 for package: device-mapper-multipath-0.4.9-
56.el6_3.1.x86_64
…

Dependencies Resolved

================================================================================================================
==============================
 Package                          Arch              Version                   Repository           Size
================================================================================================================
==============================
Updating:
 device-mapper-multipath          x86_64            0.4.9-56.el6_3.1          updates              96 k
Updating for dependencies:
 device-mapper-multipath-libs     x86_64            0.4.9-56.el6_3.1          updates              158 k
 kpartx                           x86_64            0.4.9-56.el6_3.1          updates              51 k

Transaction Summary
================================================================================================================
==============================
```

➢ パッケージインストールの例(続き)

```
Upgrade      3 Package(s)

Total download size: 305 k
Is this ok [y/N]: y
Downloading Packages:
(1/3): device-mapper-multipath-0.4.9-56.el6_3.1.x86_64.rpm                                  |  96 kB     00:00
(2/3): device-mapper-multipath-libs-0.4.9-56.el6_3.1.x86_64.rpm                             | 158 kB     00:00
(3/3): kpartx-0.4.9-56.el6_3.1.x86_64.rpm                                                   |  51 kB     00:00
--------------------------------------------------------------------------------------------------------------------------------
Total                                                       1.4 MB/s | 305 kB     00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : kpartx-0.4.9-56.el6_3.1.x86_64                                                1/6
  Updating   : device-mapper-multipath-libs-0.4.9-56.el6_3.1.x86_64                           2/6
  Updating   : device-mapper-multipath-0.4.9-56.el6_3.1.x86_64                               3/6
  Cleanup    : device-mapper-multipath-0.4.9-46.el6.x86_64                                   4/6
  Cleanup    : device-mapper-multipath-libs-0.4.9-46.el6.x86_64                              5/6
  Cleanup    : kpartx-0.4.9-46.el6.x86_64                                                    6/6

Updated:
  device-mapper-multipath.x86_64 0:0.4.9-56.el6_3.1

Dependency Updated:
  device-mapper-multipath-libs.x86_64 0:0.4.9-56.el6_3.1          kpartx.x86_64 0:0.4.9-56.el6_3.1

Complete!
$
```

➢ パッケージが複数バージョンインストールされている時の削除の例

```
$ sudo yum --installroot=/tmp/rpmtest remove kpartx
Loaded plugins: downloadonly, fastestmirror, priorities, refresh-packagekit, security
…
Dependencies Resolved

================================================================================
 Package        Arch          Version                Repository        Size
================================================================================
Removing:
 kpartx         x86_64        0.4.9-46.el6           @c6-media         38 k
 kpartx         x86_64        0.4.9-46.el6.vaj1        installed       38 k

Transaction Summary
================================================================================
Remove       2 Package(s)

Installed size: 76 k
Is this ok [y/N]: y
…
Removed:
  kpartx.x86_64 0:0.4.9-46.el6          kpartx.x86_64 0:0.4.9-46.el6.vaj1

Complete!
$ sudo rpm -qa --root /tmp/rpmtest | grep kpartx
$
```

パッケージグループの操作

機能	コマンド
パッケージグループのインストール	yum groupinstall packagegroup-name
パッケージグループの更新	yum groupupdate packagegroup-name
パッケージグループのアンインストール	yum groupremove packagegroup-name
パッケージグループの情報表示	yum groupinfo packagegroup-name
パッケージグループの一覧	yum grouplist

その他の操作

機能	コマンド
パッケージの検索	yum search search-string
インストール済みパッケージの一覧	yum list installed
インストール可能なパッケージの一覧	yum list available
アップデート可能なパッケージの一覧	yum check-update yum list updates
キャッシュ情報のクリア	yum clean all

➢パッケージグループの操作

```
$ sudo yum grouplist | grep "SNMP Support"
   SNMP Support
$ sudo yum groupinfo "SNMP Support"
Loaded plugins: downloadonly, fastestmirror, priorities, refresh-packagekit, security
...
Group: SNMP Support
 Description: SNMP management agent.
 Mandatory Packages:
   net-snmp
 Default Packages:
   net-snmp-utils
 Optional Packages:
   net-snmp-perl
   net-snmp-python
$ sudo yum groupinstall "SNMP Support"
Loaded plugins: downloadonly, fastestmirror, priorities, refresh-packagekit, security
...
Dependencies Resolved

================================================================================
 Package              Arch          Version                 Repository      Size
================================================================================
Installing:
 net-snmp             x86_64        1:5.5-41.el6_3.1        updates        302 k
 net-snmp-utils       x86_64        1:5.5-41.el6_3.1        updates        170 k
Installing for dependencies:
 net-snmp-libs        x86_64        1:5.5-41.el6_3.1        updates        1.5 M

Transaction Summary
================================================================================
Install       3 Package(s)

Total download size: 2.0 M
…
```

パッケージの自動更新

yum-cron パッケージでパッケージの自動更新が可能。

```
# yum --disablerepo=¥* --enablerepo=c6-media install yum-cron
…
# chkconfig yum-cron on
# service yum-cron start
#
```

トランザクション機能

yum実行時のトランザクション履歴が採取されている。
履歴から下記の操作が行える。

- コマンド実行日時の確認
- 実行結果の確認
- トランザクションの取り消し
- トランザクションの再実行
- 履歴情報の出力とその再実行

トランザクション情報は sqlite DB に格納されている。

トランザクションの操作コマンド

オプション	機能
履歴の一覧(start_id..end_idは範囲指定)	yum history list [pkgname | all | start_id..end_id]
履歴のサマリ表示 (start_id..end_idは範囲指定)	yum history summary [pkgname | start_id..end_id]
特定のパッケージの履歴表示	yum history package-list pkgname
履歴の詳細情報の表示	yum history info pkgname | id | start_id..end_id
追加情報の表示	yum history addon-info pkgname | id | start_id..end_id
トランザクションの取り消し	yum history undo id
トランザクションの再実行	yum history redo id
履歴の出力	yum -q history addon-info *id* saved_tx > file-name
ファイルの履歴の実行	yum load-transaction *file-name*

- 範囲指定の履歴一覧

```
$ sudo yum history list 17..20
Loaded plugins: downloadonly, fastestmirror, refresh-packagekit, security
ID     | Login user             | Date and time    | Action(s)      | Altered
-------------------------------------------------------------------------------
   20 | test <test>            | 2013-01-15 13:58 | I, U           |    3  <
   19 | test <test>            | 2013-01-15 13:53 | Downgrade      |    3 ><
   18 | test <test>            | 2013-01-15 13:40 | Reinstall      |    1 >
   17 | test <test>            | 2013-01-15 11:22 | Install        |    2
Warning: RPMDB altered outside of yum.
history list
$
```

- 履歴のサマリ表示

```
$ sudo yum history summary
Loaded plugins: downloadonly, fastestmirror, refresh-packagekit, security
Login user             | Time              | Action(s)       | Altered
-------------------------------------------------------------------------------
test <test>            | Last week         | E, I            |      14
test <test>            | Last 2 weeks      | Install         |      15
System <unset>         | Last 3 months     | Install         |     208
root <root>            | Last 3 months     | Install         |     552
test <test>            | Last 3 months     | D, E, I, R, U   |     297
history summary
$
```

➢ 範囲指定の履歴一覧

```
$ sudo yum history package-list device-mapper*
Loaded plugins: downloadonly, fastestmirror, refresh-packagekit, security
ID     | Action(s)      | Package
-------------------------------------------------------------------------------
   20 | Install        | device-mapper-multipath-0.4.9-56.el6.x86_64        <
   20 | Dep-Install    | device-mapper-multipath-libs-0.4.9-56.el6.x86_64    <
   19 | Downgrade      | device-mapper-multipath-0.4.9-46.el6.x86_64        ><
   19 | Downgraded     |                    0.4.9-56.el6_3.1.x86_64    ><
   19 | Downgrade      | device-mapper-multipath-libs-0.4.9-46.el6.x86_64   ><
   19 | Downgraded     |                        0.4.9-56.el6_3.1.x86_64 ><
   18 | Reinstall      | device-mapper-multipath-0.4.9-46.el6.x86_64        >
   15 | Erase          | device-mapper-multipath-0.4.9-56.el6_3.1.x86_64    EE
   15 | Erase          | device-mapper-multipath-libs-0.4.9-56.el6_3.1.x86_64 EE
   13 | Updated        | device-mapper-multipath-0.4.9-46.el6.x86_64
   13 | Update         |                    0.4.9-56.el6_3.1.x86_64
   13 | Updated        | device-mapper-multipath-libs-0.4.9-46.el6.x86_64
   13 | Update         |                        0.4.9-56.el6_3.1.x86_64
   11 | Install        | device-mapper-multipath-0.4.9-46.el6.x86_64
   11 | Dep-Install    | device-mapper-multipath-libs-0.4.9-46.el6.x86_64
    1 | Dep-Install    | device-mapper-1.02.66-6.el6.x86_64
    1 | Dep-Install    | device-mapper-event-1.02.66-6.el6.x86_64
    1 | Dep-Install    | device-mapper-event-libs-1.02.66-6.el6.x86_64
    1 | Dep-Install    | device-mapper-libs-1.02.66-6.el6.x86_64
Warning: RPMDB altered outside of yum.
history package-list
$
```

➢ 履歴詳細情報の表示

```
$ sudo yum history info 20
Loaded plugins: downloadonly, fastestmirror, refresh-packagekit, security
Transaction ID : 20
Begin time     : Tue Jan 15 13:58:58 2013
Begin rpmdb    : 1037:1a4d23d0a767ed5e1736be791ad08d63d2164cf4
End time       :          13:59:05 2013 (7 seconds)
End rpmdb      : 1039:46f8e4449583195d561d8c7fd4eb731598d1fb41
User           : test <test>
Return-Code    : Success
Command Line   : --disablerepo=* --enablerepo=base install device-mapper-multipath
Transaction performed with:
   Installed     rpm-4.8.0-19.el6.x86_64                  @anaconda-CentOS-201112091719.x86_64/6.2
   Installed     yum-3.2.29-22.el6.centos.noarch          @anaconda-CentOS-201112091719.x86_64/6.2
   Installed     yum-plugin-fastestmirror-1.1.30-10.el6.noarch @anaconda-CentOS-
201112091719.x86_64/6.2
Packages Altered:
   Install     device-mapper-multipath-0.4.9-56.el6.x86_64      @base
   Dep-Install device-mapper-multipath-libs-0.4.9-56.el6.x86_64 @base
   Updated     kpartx-0.4.9-46.el6.x86_64                  @c6-media
   Update          0.4.9-56.el6.x86_64                  @base
history info
$
```

➢ トランザクションの再実行

```
$ sudo yum history package-list device-mapper-multipath
Loaded plugins: downloadonly, fastestmirror, priorities, refresh-packagekit, security
ID     | Action(s)      | Package
---------------------------------------------------------------------------
    54 | Erase          | device-mapper-multipath-0.4.9-46.el6.x86_64       EE
    53 | Install        | device-mapper-multipath-0.4.9-46.el6.x86_64
    50 | Updated        | device-mapper-multipath-0.4.9-46.el6.x86_64
...
history package-list
$ rpm -qa | grep device-mapper-multipath
$ sudo yum --disablerepo=¥* --enablerepo=c6-media history redo 53
Loaded plugins: downloadonly, fastestmirror, priorities, refresh-packagekit, security
...
Dependencies Resolved

================================================================================================================
 Package                        Arch          Version               Repository       Size
================================================================================================================
Installing:
 device-mapper-multipath            x86_64        0.4.9-46.el6          c6-media         89 k
Installing for dependencies:
 device-mapper-multipath-libs       x86_64        0.4.9-46.el6          c6-media        141 k

Transaction Summary
================================================================================================================
Install       2 Package(s)
...
Installed:
  device-mapper-multipath.x86_64 0:0.4.9-46.el6

Dependency Installed:
  device-mapper-multipath-libs.x86_64 0:0.4.9-46.el6

Complete!
```

➢ トランザクションの再実行(続き)

```
$ rpm -qa | grep device-mapper-multipath
device-mapper-multipath-libs-0.4.9-46.el6.x86_64
device-mapper-multipath-0.4.9-46.el6.x86_64
$ sudo yum --disablerepo=¥* --enablerepo=c6-media history redo 54
Loaded plugins: downloadonly, fastestmirror, priorities, refresh-packagekit, security
...
Dependencies Resolved

================================================================================
 Package                Arch        Version            Repository      Size
================================================================================
Removing:
 device-mapper-multipath          x86_64      0.4.9-46.el6        @c6-media      181 k
 device-mapper-multipath-libs     x86_64      0.4.9-46.el6        @c6-media      447 k

Transaction Summary
================================================================================
Remove       2 Package(s)
...
Removed:
  device-mapper-multipath.x86_64 0:0.4.9-46.el6    device-mapper-multipath-libs.x86_64 0:0.4.9-46.el6

Complete!
$ rpm -qa | grep device-mapper-multipath
$
```

➢トランザクションの取り消し

```
$ sudo yum history list
Loaded plugins: downloadonly, fastestmirror, priorities, refresh-packagekit, security
ID     | Login user             | Date and time    | Action(s)      | Altered
-------------------------------------------------------------------------------
    57 | test <test>            | 2013-02-27 16:14 | Install        |    2
    56 | test <test>            | 2013-02-27 15:58 | Erase          |    2 EE
...
history list
$ sudo yum history undo 57
Loaded plugins: downloadonly, fastestmirror, priorities, refresh-packagekit, security
...
Dependencies Resolved

================================================================================
 Package                Arch          Version              Repository        Size
================================================================================
Removing:
 device-mapper-multipath            x86_64        0.4.9-46.el6          @c6-media        181 k
 device-mapper-multipath-libs        x86_64        0.4.9-46.el6          @c6-media        447 k

Transaction Summary
================================================================================
Remove       2 Package(s)
...
Removed:
  device-mapper-multipath.x86_64 0:0.4.9-46.el6    device-mapper-multipath-libs.x86_64 0:0.4.9-46.el6

Complete!
$ rpm -qa | grep device-mapper-multipath
$
```

➢ 履歴の出力

```
$ sudo yum -q history addon-info 20 saved_tx
1037:1a4d23d0a767ed5e1736be791ad08d63d2164cf4
0
1
base:6346:1341568627
4
mbr: device-mapper-multipath,x86_64,0,0.4.9,56.el6 70
  repo: base
  ts_state: u
  output_state: 20
  isDep: False
  reason: user
  reinstall: False
mbr: kpartx,x86_64,0,0.4.9,46.el6 20
  repo: installed
  ts_state: ud
  output_state: 90
  isDep: False
  reason: user
  reinstall: False
  relatedto: kpartx,x86_64,0,0.4.9,56.el6@a:updatedby
  updated_by: kpartx,x86_64,0,0.4.9,56.el6@a
mbr: kpartx,x86_64,0,0.4.9,56.el6 70
…
```

パッケージのビルド

パッケージ名

package-name-{version}-{release}.src.rpm

package-name-{version}-{release}.{arch}rpm

ビルドに必要なもの

- 該当パッケージのソースパッケージ

  下記サイトからダウンロード可。
  http://vault.centos.org/6.2/os/Source/SPackages/

- ビルド関連ツール

  rpm-build 等のパッケージが必要となるが通常 “Development tools” パッケージグループをインストールすれば良い

- 対象パッケージ固有のビルド依存パッケージ

  spec ファイルのBuildRequiresに指定のパッケージ

  yum-builddep を使うと便利。

yum-builddep の使用例

```
$ sudo yum-builddep --disablerepo=¥* --enablerepo=c6-media device-mapper-multipath-0.4.9-46.el6.src.rpm
Loaded plugins: fastestmirror, priorities, refresh-packagekit
…
c6-media                                  | 4.0 kB     00:00 ...
Checking for new repos for mirrors
Getting requirements for device-mapper-multipath-0.4.9-46.el6.src
 --> Already installed : libaio-devel-0.3.107-10.el6.x86_64
 --> device-mapper-devel-1.02.66-6.el6.x86_64
 --> Already installed : libselinux-devel-2.0.94-5.2.el6.x86_64
 --> Already installed : libsepol-devel-2.0.41-4.el6.x86_64
 --> Already installed : readline-devel-6.0-3.el6.x86_64
 --> Already installed : ncurses-devel-5.7-3.20090208.el6.x86_64
--> Running transaction check
---> Package device-mapper-devel.x86_64 0:1.02.66-6.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package              Arch        Version           Repository      Size
================================================================================
Installing:
 device-mapper-devel      x86_64      1.02.66-6.el6      c6-media       82 k
```

yum-builddep の使用例(続き)

```
Transaction Summary
================================================================================
Install       1 Package(s)

Total download size: 82 k
Installed size: 41 k
Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
Warning: RPMDB altered outside of yum.
  Installing : device-mapper-devel-1.02.66-6.el6.x86_64                    1/1

Installed:
  device-mapper-devel.x86_64 0:1.02.66-6.el6

Complete!
$
```

再構築を行うだけの場合

- ソースパッケージを使用する場合

```
$ rpmbuild --rebuild source-package-file-name
```

- tarball を使用する場合

```
$ rpmbuild -ta tarball-name
```

- ビルドの場所

  RHEL6以降 : ~/rpmbuild

  RHEL5以前 : /usr/src/redhat

パッケージをカスタマイズする場合

ビルドの流れ

- ソースパッケージのインストール
- パッチファイル作成
- 必要ならばspecファイル修正
- ビルド実行

予備知識

パッケージ構築の為の環境

- マクロ
  - デフォルトのマクロ: “rpmbuild --showrc”
  - ユーザ定義:~/. rpmmacros

```
 $ rpm --showrc |grep _topdir
…
-14: _topdir    %{getenv:HOME}/rpmbuild
$ echo "%_topdir /home/test/packages/build" > ~/.rpmmacros
$ rpm --showrc |grep _topdir
…
-14: _topdir    /home/test/packages/build
$
```

- 引数でのマクロ指定

  --define ‘マクロ名 変更値’

➢ビルドに関係するマクロ(CentOS6.2)

```
$ rpmbuild --showrc
...
-14: _build_vendor     %{_host_vendor}
-14: _builddir  %{_topdir}/BUILD
-14: _buildrootdir     %{_topdir}/BUILDROOT
…
-14: _rpmdir    %{_topdir}/RPMS
-14: _rpmfilename       %{_build_name_fmt}
-14: _rpmlock_path      %{_dbpath}/.rpm.lock
-14: _sbindir   %{_exec_prefix}/sbin
-14: _sharedstatedir    /var/lib
-14: _signature gpg
…
-11: _target    x86_64-linux
-14: _target_alias      %{_host_alias}
-11= _target_cpu       x86_64
-11= _target_os linux
-14: _target_platform   %{_target_cpu}-%{_vendor}-%{_target_os}%{?_gnu}
-14: _target_vendor     %{_host_vendor}
-14: _tmppath   %{_var}/tmp
-14: _topdir    %{getenv:HOME}/rpmbuild
…
-14: buildroot  %{_buildrootdir}/%{name}-%{version}-%{release}.%{_arch}
-14: centos     6
…
$
```

➢ビルドに関係するマクロ(CentOS5.3)

CentOS5.3 の時の _topdir

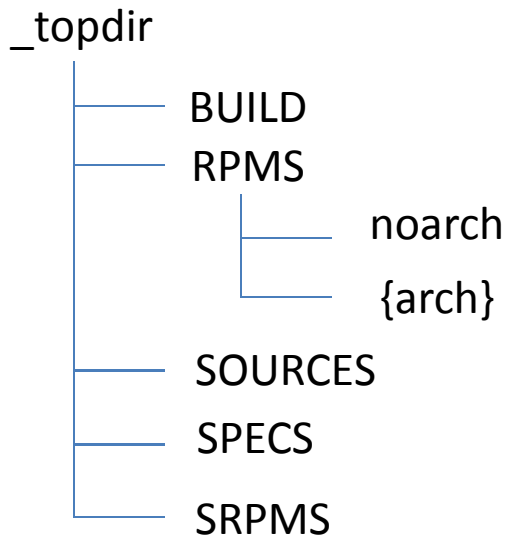
%{_usrsrc}/redhat --> “/usr/src/redhat”

```
$ 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
$
```

ソースパッケージのインストール

```
$ rpm -ivh source-package-file
```

- ビルドツリー



インストール時はSPECS, SOURCES。
それ以外はビルド時に作成。

ソースパッケージのインストール（続き）

- BUILD：ビルドの為の作業ディレクトリとして使用。
- RPMS：バイナリパッケージの置き場。アーキテクチャのディレクトリ名の下に置かれる。
- SPECS ：パッケージビルドの処理手順が書かれた spec ファイルが展開される。
- SOURCES：該当のソフトのソースコード、パッチファイルが展開される。通常カスタマイズする場合の修正もパッチファイルの形で此処に追加する。
- SRPMS:ソースパッケージの置き場

パッチファイルの作成

- ソースコードの修正は通常パッチとして作成。
- パッチはSOURCES に置く
- 複数パッチを置いても良い。

specファイルとは

- ビルド作業の手順書

  前述の様にコンパイル、パッチの適用、インストール、インストール前処理、後処理等の処理手順が書かれている。
- 各処理はセクション毎に記述。%セクション名で指定する。

specファイルの構成

specファイルは大きく下記のセクションで構成される。

- introduction
- %prep
- %build
- %install
- %clean
- %files
- %changelog

主なセクション

- introduction：specファイルの先頭から%prepセクションまでのパッケージの情報を記述。

  インストール時の依存情報の指定、ビルド時の依存情報の指定

- %prep:ビルドの準備処理。
- %build: ビルド作業処理。

  ソフトウェア本体の生成作業の実行(make の実行等)。

- %install: 必要ファイルのインストール。

  buildroot で指定のディレクトリに仮インストールする。

  --buildroot 引数で変更できる。

主なセクション（続き）

- %clean: ビルド後の後処理。
  作業ファイル等の削除を行う。
- %files: パッケージに格納するファイルを指定する。
- スクリプトのセクション
  実行タイミングにより種類が分かれている
  - %pre:パッケージインストール前に実行する処理。
  - %post: パッケージインストール後に実行する処理。
  - %preun:パッケージアンインストール前に実行する処理。
  - %postun:パッケージアンインストール後に実行する処理。
  - % toriggerin:指定のパッケージをインストールする前に処理
  - % toriggerun:指定のパッケージをアンインストールする前に処理
  - % toriggerpostun:指定のパッケージをアンインストールした後に処理
  - % toriggerin:指定のパッケージをインストールする前に処理
- %changelog: パッケージの変更点の説明。

主なセクション（続き）

- %package: サブパッケージの情報の記述。

  複数のパッケージを作成する場合に introductionセクションで記述する。

  device-mapper-multipath パッケージでは

  %package -n kpartx -> kpartx-0.4.9-46.el6.x86_64.rpm

  %package kpartx -> device-mapper-multipath-kpartx-0.4.9-46.el6.x86_64.rpm

  サブパッケージ固有の処理は下記のセクションでも行える。

  パッケージ名の指定は上記と同様。

  %description, %files, %post, %preun

主なマクロ

- %setup [options]: ソースアーカイブの展開。

  introduction の Source または Source0タグに指定されたアーカイブを展開し、そのディレクトリへ位置付ける。

  -a 番号: ディレクトリ移動後に番号のソースを展開。

  -b 番号: ディレクトリ移動前に番号のソースを展開。

  -c: 展開前にディレクトリを作成。アーカイブでディレクトリが作成されない場合に使用。

  -D: 展開前にディレクトリを削除しない。

  -n directory-name: アーカイブで展開されるディレクトリ名

  -q: メッセージを表示しない。

  -T: Source または Source0タグのアーカイブを展開しない。

  複数のソースを展開する場合は下記の様に指定。

  ```
  %setup
  %setup -T -D -a 1
  ```

主なマクロ（続き）

- %patch番号: パッチに適用。

  Patch番号: で指定されたパッチを適用する。

  -p 数値: 数値で指定された分のディレクトリを除く。

  -b 拡張子: パッチ元のファイルを拡張子名でセーブする。

定義済み変数

マクロで設定された定義済みの環境変数があり、スクリプトなどで参照できる。

変数名	内容
RPM_SOURCE_DIR	SOURCES ディレクトリパス
RPM_BUILD_DIR	BUILDディレクトリパス
RPM_DOC_DIR	ドキュメントインストール先のディレクトリパス (/usr/share/doc)
RPM_OPT_FLAGS	コンパイルオプション
RPM_ARCH	システムのアーキテクチャ(x86_64等)
RPM_OS	システムのOS名(linux等)
RPM_PACKAGE_NAME	パッケージ名(introduction のNameタグ)
RPM_PACKAGE_VERSION	パッケージ名(introduction のVersionタグ)
RPM_PACKAGE_RELEASE	パッケージ名(introduction のReleaseタグ)
RPM_BUILD_ROOT	仮インストール先ディレクトリ(buildroot)

➢定義済み変数例

```
$ rpm --showrc | grep RPM_PACKAGE_
  RPM_PACKAGE_NAME="%{name}"
  RPM_PACKAGE_VERSION="%{version}"
  RPM_PACKAGE_RELEASE="%{release}"
  export RPM_PACKAGE_NAME RPM_PACKAGE_VERSION RPM_PACKAGE_RELEASE
$
```

➢specファイル例

```
Summary: Tools to manage multipath devices using device-mapper
Name: device-mapper-multipath
Version: 0.4.9
Release: 46%{?dist}
License: GPL+
Group: System Environment/Base
URL: http://christophe.varoqui.free.fr/

Source0: multipath-tools-091027.tar.gz                    --> アーカイブファイルの指定
Source1: multipath.conf
# patch that should go upstream
Patch1: 0001-for-upstream-add-tpg_pref-prioritizer.patch  --> パッチファイルの指定
...
Patch1118: 0118-RHBZ-738298-revert-631009.patch


# runtime
Requires: %{name}-libs = %{version}-%{release}
Requires: kpartx = %{version}-%{release}
Requires: device-mapper >= 1.02.38
Requires(post): chkconfig
Requires(preun): chkconfig
Requires(preun): initscripts
Requires(postun): initscripts
```

➢specファイル例

```
# build/setup
BuildRequires: libaio-devel, device-mapper-devel >= 1.02.38
BuildRequires: libselinux-devel, libsepol-devel
BuildRequires: readline-devel, ncurses-devel

BuildRoot: %(mktemp -ud %{_tmppath}/%{name}-%{version}-%{release}-XXXXXX)

%description                                          --> パッケージの説明
%{name} provides tools to manage multipath devices by
instructing the device-mapper multipath kernel module what to do.
The tools are :
* multipath - Scan the system for multipath devices and assemble them.
* multipathd - Detects when paths fail and execs multipath to update things.

%package libs                                         --> サブパッケージ定義
Summary: The %{name} modules and shared library
License: GPL+
Group: System Environment/Libraries

%description libs                                     --> サブパッケージの説明
The %{name}-libs provides the path checker
and prioritizer modules. It also contains the multipath shared library,
libmultipath.
```

➢specファイル例

```
%package -n kpartx                                          --> サブパッケージ定義(名前指定)
Summary: Partition device manager for device-mapper devices
Group: System Environment/Base

%description -n kpartx
kpartx manages partition creation and removal for device-mapper devices.

%prep                                                       --> ビルドの準備作業
%setup -q -n multipath-tools
%patch1 -p1
...
%patch1118 -p1
cp %{SOURCE1} .

%build                                                      --> プログラムのビルド
%define _sbindir /sbin
%define _libdir /%{_lib}
%define _libmpathdir %{_libdir}/multipath
make %{?_smp_mflags} LIB=%{_lib}
```

➢specファイル例

```
%install                                                    --> buildrootへの仮インストール
rm -rf %{buildroot}

make install ¥
      DESTDIR=%{buildroot} ¥
      bindir=%{_sbindir} ¥
      syslibdir=%{_libdir} ¥
      libdir=%{_libmpathdir} ¥
      rcdir=%{_initrddir}

# tree fix up
# install -m 0644 %{SOURCE1} %{buildroot}/etc/multipath.conf
install -d %{buildroot}/etc/multipath

%clean
rm -rf %{buildroot}

%post                                                       --> インストール後の処理
/sbin/chkconfig --add multipathd
if [ "$1" -gt "1" -a ! -e /etc/multipath/bindings -a ¥                  注1)
   -f /var/lib/multipath/bindings ]; then
      mv /var/lib/multipath/bindings /etc/multipath/bindings
      ln -s /etc/multipath/bindings /var/lib/multipath/bindings
fi
```

➢ specファイル例

```
%preun                                                          --> アンインストール前の処理
if [ “$1” = 0 ]; then                                                   注1)
      /sbin/service multipathd stop /dev/null 2>&1
      /sbin/chkconfig --del multipathd
fi

%postun                                                         --> アンインストール前の処理
if [ "$1" -ge "1" ]; then                                               注1)
      /sbin/service multipathd condrestart >/dev/null 2>&1 || :
fi

%files                                                          --> パッケージに含めるファイル
%defattr(-,root,root,-)                                         --> ファイルのモード、オーナ、グループ指定
%{_sbindir}/multipath                                                   注2)
%{_sbindir}/multipathd
%{_sbindir}/cciss_id
%{_sbindir}/mpathconf
%{_initrddir}/multipathd
%{_mandir}/man5/multipath.conf.5.gz
%{_mandir}/man8/multipath.8.gz
%{_mandir}/man8/multipathd.8.gz
%{_mandir}/man8/mpathconf.8.gz
%config /lib/udev/rules.d/40-multipath.rules                            注3)
%doc AUTHOR COPYING FAQ
%doc multipath.conf multipath.conf.annotated
%doc multipath.conf.defaults multipath.conf.synthetic
%dir /etc/multipath
```

➢specファイル例

```
%files libs                                                    --> サブパッケージに含めるファイル
%defattr(-,root,root,-)
%doc AUTHOR COPYING
%{_libdir}/libmultipath.so
%dir %{_libmpathdir}
%{_libmpathdir}/*

%post libs -p /sbin/ldconfig

%postun libs -p /sbin/ldconfig

%files -n kpartx                                       --> サブパッケージに含めるファイル(名前指定)
%defattr(-,root,root,-)
/sbin/kpartx
%{_mandir}/man8/kpartx.8.gz

%changelog
* Wed Oct  5 2011 Benjamin Marzinski <bmarzins@redhat.com> -0.4.9.46
- Remove 0107-RHBZ-725541-01-async-tur-checker.patch
  * This patch caused a regression where, multpathd could corrupt is memory
    when a path was deleted (bz747604)
…
```

注1)　%pre, post, %preun, %postun セクション実行時にインストールの状態値が $1 で渡って来る。その値を参照する事でどういう操作で実行されているかがわかる。

0 : rpm -e でアンインストールする時。

1 : 初回インストール

2以上 : アップグレード時

rpm -U 実行時の動作は新しいパッケージインストール後、古いパッケージがアンインストールされる。そこでセクションは %pre, %post, %preun, %postun の順に実行される事になる。その為、上記の様に $1 = 0 の判定をしないと、新しいパッケージでインストールしたファイルが古いパッケージのアンインストール時に操作されてしまうので注意が必要。

注2)　システムのパス用にマクロが用意されている。

_sysconfdir: /etc, _usr: /usr, _var: /var

_sbindir: /usr/sbin , _bindir: /usr/bin, _usrsrc: /usr/src

例: %{_sbindir}/cciss_id

またワイルドカードも使用できる。

例: %{_sbindir}/*

注3)　設定ファイルの指定。既存のファイルは置き換えられ xxx.rpmsave にリネームされる。

%config(noreplace) file-name 書式の場合既存のファイルは置き換えられず、インストールファイルが xxx.rpmnew というファイル名で置かれる。

ビルド実行

- rpmbuild [options] specfile-path
  - options

    -bp: ソース展開、パッチ適用まで

    -bb: バイナリパッケージ作成まで

    -bs: ソースパッケージ作成まで

    -ba: バイナリ、ソースパッケージ作成

    -bi:インストールまで

    前述 --define, --buildroot 等。

➢ビルド例

```
$ mkdir -p /tmp/device-mapper-multipath-0.4.9-46.vaj.el6.vaj1
$ rpm -ivh --define '_topdir /tmp/device-mapper-multipath-0.4.9-46.el6.vaj1' device-mapper-multipath-
0.4.9-46.el6.src.rpm
 1:device-mapper-multipathwarning: user mockbuild does not exist - using root
...
$ ls /tmp/device-mapper-multipath-0.4.9-46.el6.vaj1
SOURCES  SPECS
$ diff -u SPECS/device-mapper-multipath.spec.orig SPECS/device-mapper-multipath.spec
--- SPECS/device-mapper-multipath.spec.orig     2011-10-26 05:57:54.000000000 +0900
+++ SPECS/device-mapper-multipath.spec  2013-02-24 21:48:05.140034670 +0900
@@ -6,6 +6,8 @@
 Group: System Environment/Base
 URL: http://christophe.varoqui.free.fr/

+Prefix:/sbin
+
 Source0: multipath-tools-091027.tar.gz
 Source1: multipath.conf
 # patch that should go upstream
$ rpmbuild -ba --define '_topdir /tmp/device-mapper-multipath-0.4.9-46.el6.vaj1' --define 'dist .el6.vaj1'
/tmp/device-mapper-multipath-0.4.9-46.el6.vaj1/SPECS/device-mapper-multipath.spec
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.f2N20w
...
+ exit 0
$ ls /tmp/device-mapper-multipath-0.4.9-46.el6.vaj1/
BUILD  BUILDROOT  RPMS  SOURCES  SPECS  SRPMS
```

➢ビルド例（続き）

```
$ ls /tmp/device-mapper-multipath-0.4.9-46.el6.vaj1/SRPMS/
device-mapper-multipath-0.4.9-46.el6.vaj1.src.rpm
$ ls /tmp/device-mapper-multipath-0.4.9-46.el6.vaj1/RPMS/x86_64/
device-mapper-multipath-0.4.9-46.el6.vaj1.x86_64.rpm
device-mapper-multipath-debuginfo-0.4.9-46.el6.vaj1.x86_64.rpm
device-mapper-multipath-libs-0.4.9-46.el6.vaj1.x86_64.rpm
kpartx-0.4.9-46.el6.vaj1.x86_64.rpm
$ rpm -qa | egrep 'kpartx|device-mapper-multi'
device-mapper-multipath-libs-0.4.9-46.el6.x86_64
kpartx-0.4.9-46.el6.x86_64
device-mapper-multipath-0.4.9-46.el6.x86_64
$ sudo rpm -Uvh --replacepkgs --prefix /usr/local/sbin device-mapper-multipath-0.4.9-46.el6.x86_64.rpm
error: package device-mapper-multipath is not relocatable
$ sudo rpm -Uvh --prefix /usr/local/sbin device-mapper-multipath-0.4.9-46.el6.vaj1.x86_64.rpm device-
mapper-multipath-libs-0.4.9-46.el6.vaj1.x86_64.rpm kpartx-0.4.9-46.el6.vaj1.x86_64.rpm
Preparing...                ########################################### [100%]
   1:kpartx                 ########################################### [ 33%]
   2:device-mapper-multipath########################################### [ 67%]
   3:device-mapper-multipath########################################### [100%]
$ rpm -qa | egrep 'kpartx|device-mapper-multi'
device-mapper-multipath-libs-0.4.9-46.el6.vaj1.x86_64
kpartx-0.4.9-46.el6.vaj1.x86_64
device-mapper-multipath-0.4.9-46.el6.vaj1.x86_64
$ ls /usr/local/sbin/
cciss_id  kpartx  mpathconf  multipath  multipathd
$
```

カーネルモジュールパッケージ

- どういう場合に必要か
  - カーネルツリー内のモジュールのカスタマイズ
  - カーネルツリー内に無いモジュールの追加
- モジュールパッケージ使用時の問題点
  - 通常カーネルのバージョンに依存する為、バイナリ互換のある場合でもバージョンアップ毎にモジュールを作成する必要あり。
  - そして稼働システムのカーネルバージョンアップの際、モジュールもインストールしなければいけない。
- 回避する為の主な方式
  - DKMS（Dynamic Kernel Module Support）

    ソースビルド方式

    対象システムにdkmsコマンド、コンパイル環境が必要
  - kmod（kernel module rpm package）

    モジュール使いまわし方式。

    基本的には標準のシステムで使用できる。

カーネルモジュールパッケージ（続き）

- RedHatの方式
  - kmod方式を採用。
  - DKMS コマンドは標準で提供されていない
  - kmod 関連ツール
    - /usr/lib/rpm/redhat/kmodtool
      （ redhat-rpm-config-9.0.3-34.el6.noarch ）
    - /sbin/weak-modules
      （module-init-tools-3.9-17.el6.x86_64）
  - モジュールのリンクを操作するタイミング。
    モジュールインストール、アンインストール時。
    カーネルインストール、アンインストール時。

➢ kmod パッケージ specファイル例

```
# Define the kmod package name here.
%define kmod_name jfs

# If kversion isn't defined on the rpmbuild line, define it here.
%{!?kversion: %define kversion 2.6.32-71.el6.%{_target_cpu}}

Name:    %{kmod_name}-kmod
Version: 0.0
Release: 1%{?dist}
Group:   System Environment/Kernel
License: GPLv2
Summary: %{kmod_name} kernel module(s)
URL:     http://www.kernel.org/

BuildRequires: redhat-rpm-config
ExclusiveArch: i686 x86_64

# Sources.
Source0:  %{kmod_name}-%{version}.tar.bz2
Source5:  GPL-v2.0.txt
Source10: kmodtool-%{kmod_name}-el6.sh

# Magic hidden here.
%{expand:%(sh %{SOURCE10} rpmtemplate %{kmod_name} %{kversion} "")}          --> kmodtoolの展開

# Disable the building of the debug package(s).
%define debug_package %{nil}

%description
This package provides the %{kmod_name} kernel module(s).
It is built to depend upon the specific ABI provided by a range of releases
of the same variant of the Linux kernel and not on any one specific build.
```

➢ kmod パッケージ specファイル例

```
%prep
%setup -q -n %{kmod_name}-%{version}
%{__cp} -a %{SOURCE5} .
echo "override %{kmod_name} * weak-updates/%{kmod_name}" > kmod-%{kmod_name}.conf

%build
KSRC=%{_usrsrc}/kernels/%{kversion}
%{__make} -C "${KSRC}" %{?_smp_mflags} modules M=$PWD

%install
export INSTALL_MOD_PATH=%{buildroot}
export INSTALL_MOD_DIR=extra/%{kmod_name}
KSRC=%{_usrsrc}/kernels/%{kversion}
%{__make} -C "${KSRC}" modules_install M=$PWD
%{__install} -d %{buildroot}%{_sysconfdir}/depmod.d/
%{__install} kmod-%{kmod_name}.conf %{buildroot}%{_sysconfdir}/depmod.d/
%{__install} -d %{buildroot}%{_defaultdocdir}/kmod-%{kmod_name}-%{version}/
%{__install} GPL-v2.0.txt %{buildroot}%{_defaultdocdir}/kmod-%{kmod_name}-%{version}/
# Set the module(s) to be executable, so that they will be stripped when packaged.
find %{buildroot} -type f -name ¥*.ko -exec %{__chmod} u+x ¥{¥} ¥;
# Remove the unrequired files.
%{__rm} -f %{buildroot}/lib/modules/%{kversion}/modules.*

%clean
%{__rm} -rf %{buildroot}

%changelog
* Thu Feb 24 2011 Alan Bartlett <ajb@elrepo.org> - 0.0-1
- Initial el6 build of the kmod package.
```

kmod パッケージ specファイルの説明

- introduction 、ビルド、インストールセクションが定義されている。
- カーネルモジュール自体の定義は kmodtool がサブパッケージとして展開する。
- introductionのパッケージは %files 定義が無い為パッケージとしては生成され無い。

➢ kmodtool 例

```
#!/bin/bash

# kmodtool - Helper script for building kernel module RPMs
#          An original version appeared in Fedora. This version is
#          generally called only by the %kernel_module_package RPM macro
#          during the process of building Driver Update Packages (which
#          are also known as "kmods" in the Fedora community).
#
# Copyright (c) 2003-2010 Ville Skyttä <ville.skytta@iki.fi>,
#                         Thorsten Leemhuis <fedora@leemhuis.info>
#                         Jon Masters <jcm@redhat.com>
#
# Permission is hereby granted, free of charge, to any person obtaining
# a copy of this software and associated documentation files (the
# "Software"), to deal in the Software without restriction, including
# without limitation the rights to use, copy, modify, merge, publish,
# distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so, subject to
# the following conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
# MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
# LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
# WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

➢ kmodtool 例（続き）

```
# Changelog:
#
#       2010/07/28 - Add fixes for filelists in line with LF standard
#                  - Remove now defunct "framepointer" kernel variant
#                  - Change version to "rhel6-rh2" as a consequence.
#
#       2010/01/10 - Simplified for RHEL6. We are working on upstream
#                    moving to a newer format and in any case do not
#                    need to retain support for really old systems.

shopt -s extglob

myprog="kmodtool"
myver="rhel6-rh2"
knownvariants=@(debug|kdump)
kmod_name=
kver=
verrel=
variant=

get_verrel ()
{
  verrel=${1:-$(uname -r)}
  verrel=${verrel%%$knownvariants}
}

print_verrel ()
{
  get_verrel $@
  echo "${verrel}"
}
```

➢ kmodtool 例（続き）

```
get_variant ()
{
  get_verrel $@
  variant=${1:-$(uname -r)}
  variant=${variant##$verrel}
  variant=${variant:-""}
}

print_variant ()
{
  get_variant $@
  echo "${variant}"
}

get_filelist() {
        local IFS=$'¥n'
        filelist=($(cat))

        if [ ${#filelist[@]} -gt 0 ];
        then
                for ((n = 0; n < ${#filelist[@]}; n++));
                do
                        line="${filelist[n]}"
                        line=$(echo "$line" ¥
                                | sed -e "s/%verrel/$verrel/g" ¥
                                | sed -e "s/%variant/$variant/g" ¥
                                | sed -e "s/%dashvariant/$dashvariant/g" ¥
                                | sed -e "s/%dotvariant/$dotvariant/g" ¥
                                | sed -e "s/¥.%1/$dotvariant/g" ¥
                                | sed -e "s/¥-%1/$dotvariant/g" ¥
                                | sed -e "s/%2/$verrel/g")
                        echo "$line"
                done
```

➢ kmodtool 例（続き）

```
else
        echo "%defattr(644,root,root,755)"
        echo "/lib/modules/${verrel}${dotvariant}"
    fi
}

get_rpmtemplate ()
{
  local variant="${1}"
  local dashvariant="${variant:+-${variant}}"
  local dotvariant="${variant:+.${variant}}"

  echo "%package      -n kmod-${kmod_name}${dashvariant}"

  if [ -z "$kmod_provides_summary" ]; then
    echo "Summary:        ${kmod_name} kernel module(s)"
  fi

  if [ -z "$kmod_provides_group" ]; then
    echo "Group:          System Environment/Kernel"
  fi

  if [ ! -z "$kmod_version" ]; then
    echo "Version: %{kmod_version}"
  fi

  if [ ! -z "$kmod_release" ]; then
    echo "Release: %{kmod_release}"
  fi

  # Turn off the internal dep generator so we will use the kmod scripts.
  echo "%global _use_internal_dependency_generator 0“
```

➢ kmodtool 例（続き）

```
cat <<EOF
Provides:       kabi-modules = ${verrel}${dotvariant}
Provides:       ${kmod_name}-kmod = %{?epoch:%{epoch}:}%{version}-%{release}
Requires(post):   /sbin/depmod
Requires(postun): /sbin/depmod
EOF

   if [ "no" != "$nobuildreqs" ]
   then
      echo "BuildRequires: kernel${dashvariant}-devel"
   fi

   if [ "" != "$override_preamble" ]
   then
      cat "$override_preamble"
   fi

cat <<EOF
%description   -n kmod-${kmod_name}${dashvariant}
This package provides the ${kmod_name} kernel module(s) built
for the Linux kernel using the %{_target_cpu} family of processors.
EOF

###############################################################################
## The following are not part of this script directly, they are scripts     ##
## that will be executed by RPM during various stages of package processing ##
###############################################################################

cat <<EOF
%post         -n kmod-${kmod_name}${dashvariant}
echo "Working. This may take some time ..."
if [ -e "/boot/System.map-${verrel}${dotvariant}" ]; then
   /sbin/depmod -aeF "/boot/System.map-${verrel}${dotvariant}" "${verrel}${dotvariant}" > /dev/null || :
fi
```

➢ kmodtool 例（続き）

```
modules=( ¥$(find /lib/modules/${verrel}${dotvariant}/extra/${kmod_name} | grep '¥.ko$') )
if [ -x "/sbin/weak-modules" ]; then
   printf '%s¥n' "¥${modules[@]}" | /sbin/weak-modules --add-modules
fi
echo "Done."
EOF

cat <<EOF
%preun        -n kmod-${kmod_name}${dashvariant}
rpm -ql kmod-${kmod_name}${dashvariant}-%{version}-%{release}.$(arch) | grep '¥.ko$' > /var/run/rpm-kmod-
${kmod_name}${dashvariant}-modules
EOF

cat <<EOF
%postun       -n kmod-${kmod_name}${dashvariant}
echo "Working. This may take some time ..."
if [ -e "/boot/System.map-${verrel}${dotvariant}" ]; then
   /sbin/depmod -aeF "/boot/System.map-${verrel}${dotvariant}" "${verrel}${dotvariant}" > /dev/null || :
fi
modules=( ¥$(cat /var/run/rpm-kmod-${kmod_name}${dashvariant}-modules) )
rm /var/run/rpm-kmod-${kmod_name}${dashvariant}-modules
if [ -x "/sbin/weak-modules" ]; then
   printf '%s¥n' "¥${modules[@]}" | /sbin/weak-modules --remove-modules
fi
echo "Done."
EOF

echo "%files        -n kmod-${kmod_name}${dashvariant}"
if [ "" == "$override_filelist" ];
then
   echo "%defattr(644,root,root,755)"
   echo "/lib/modules/${verrel}${dotvariant}/”
```

➢ kmodtool 例（続き）

```
    echo "%config /etc/depmod.d/kmod-${kmod_name}.conf"
    echo "%doc /usr/share/doc/kmod-${kmod_name}-%{version}/"
else
    cat "$override_filelist" | get_filelist
fi
}

print_rpmtemplate ()
{
  kmod_name="${1}"
  shift
  kver="${1}"
  get_verrel "${1}"
  shift
  if [ -z "${kmod_name}" ] ; then
    echo "Please provide the kmodule-name as first parameter." >&2
    exit 2
  elif [ -z "${kver}" ] ; then
    echo "Please provide the kver as second parameter." >&2
    exit 2
  elif [ -z "${verrel}" ] ; then
    echo "Couldn't find out the verrel." >&2
    exit 2
  fi

  for variant in "$@" ; do
      if [ "default" == "$variant" ];
      then
          get_rpmtemplate ""
      else
          get_rpmtemplate "${variant}"
      fi
  done
}
```

➢ kmodtool 例（続き）

```
usage ()
{
  cat <<EOF
You called: ${invocation}

Usage: ${myprog} <command> <option>+
 Commands:
  verrel <uname>
    - Get "base" version-release.
  variant <uname>
    - Get variant from uname.
  rpmtemplate <mainpgkname> <uname> <variants>
    - Return a template for use in a source RPM
  version
    - Output version number and exit.
EOF
}

invocation="$(basename ${0}) $@"
while [ "${1}" ] ; do
  case "${1}" in
    verrel)
      shift
      print_verrel $@
      exit $?
      ;;
    variant)
      shift
      print_variant $@
      exit $?
      ;;
```

➢ kmodtool 例（続き）

```
rpmtemplate)
     shift
     print_rpmtemplate "$@"
     exit $?
     ;;
   version)
     echo "${myprog} ${myver}"
     exit 0
     ;;
   *)
     echo "Error: Unknown option '${1}'." >&2
     usage >&2
     exit 2
     ;;
  esac
done

# Local variables:
# mode: sh
# sh-indentation: 2
# indent-tabs-mode: nil
# End:
# ex: ts=2 sw=2 et
```

➢ kmodtool 実行イメージ

```
$ ./kmodtool-jfs-el6.sh rpmtemplate jfs 2.6.32-71.el6.x86_64 ""
%package      -n kmod-jfs
Summary:        jfs kernel module(s)
Group:          System Environment/Kernel
%global _use_internal_dependency_generator 0
Provides:       kabi-modules = 2.6.32-71.el6.x86_64
Provides:       jfs-kmod = %{?epoch:%{epoch}:}%{version}-%{release}
Requires(post):   /sbin/depmod
Requires(postun): /sbin/depmod
BuildRequires: kernel-devel
%description   -n kmod-jfs
This package provides the jfs kernel module(s) built
for the Linux kernel using the %{_target_cpu} family of processors.
%post         -n kmod-jfs
echo "Working. This may take some time ..."
if [ -e "/boot/System.map-2.6.32-71.el6.x86_64" ]; then
   /sbin/depmod -aeF "/boot/System.map-2.6.32-71.el6.x86_64" "2.6.32-71.el6.x86_64" > /dev/null || :
fi
modules=( $(find /lib/modules/2.6.32-71.el6.x86_64/extra/jfs | grep '¥.ko$') )
if [ -x "/sbin/weak-modules" ]; then
   printf '%s¥n' "${modules[@]}" | /sbin/weak-modules --add-modules
fi
echo "Done."
%preun        -n kmod-jfs
rpm -ql kmod-jfs-%{version}-%{release}.x86_64 | grep '¥.ko$' > /var/run/rpm-kmod-jfs-modules
%postun       -n kmod-jfs
echo "Working. This may take some time ...”
```

➢ kmodtool 実行イメージ(続き)

```
if [ -e "/boot/System.map-2.6.32-71.el6.x86_64" ]; then
   /sbin/depmod -aeF "/boot/System.map-2.6.32-71.el6.x86_64" "2.6.32-71.el6.x86_64" > /dev/null || :
fi
modules=( $(cat /var/run/rpm-kmod-jfs-modules) )
rm /var/run/rpm-kmod-jfs-modules
if [ -x "/sbin/weak-modules" ]; then
   printf '%s¥n' "${modules[@]}" | /sbin/weak-modules --remove-modules
fi
echo "Done."
%files        -n kmod-jfs
%defattr(644,root,root,755)
/lib/modules/2.6.32-71.el6.x86_64/
%config /etc/depmod.d/kmod-jfs.conf
%doc /usr/share/doc/kmod-jfs-%{version}/
$
```

パッケージのテスト環境の構築

- こう言う時に便利
  - システム環境を壊したく無い場合
  - 特定の条件の環境を設定したい場合
- 作成の流れ
  - ディレクトリ作成とrpmデータベースの作成
  - 基本パッケージのインストール

    setup, basesystem, glibc, lvm2-libs

    yum を使うのが便利

➢ パッケージテスト環境の構築例

```
$ mkdir -p /tmp/rpmtest/var/lib/rpm
$ sudo rpm -ivh --root /tmp/rpmtest2 kpartx-0.4.9-46.el6.x86_64.rpm
warning: kpartx-0.4.9-46.el6.x86_64.rpm: Header V3 RSA/SHA1 Signature, key ID c105b9de: NOKEY
error: Failed dependencies:
      libc.so.6()(64bit) is needed by kpartx-0.4.9-46.el6.x86_64
      libc.so.6(GLIBC_2.2.5)(64bit) is needed by kpartx-0.4.9-46.el6.x86_64
      libc.so.6(GLIBC_2.3)(64bit) is needed by kpartx-0.4.9-46.el6.x86_64
      libc.so.6(GLIBC_2.3.3)(64bit) is needed by kpartx-0.4.9-46.el6.x86_64
      libdevmapper.so.1.02()(64bit) is needed by kpartx-0.4.9-46.el6.x86_64
      libdevmapper.so.1.02(Base)(64bit) is needed by kpartx-0.4.9-46.el6.x86_64
      rtld(GNU_HASH) is needed by kpartx-0.4.9-46.el6.x86_64
$ rpm --initdb --root /tmp/rpmtest
$ sudo yum --disablerepo=¥* --enablerepo=c6-media install --installroot=/tmp/rpmtest setup
$ sudo yum --disablerepo=¥* --enablerepo=c6-media install --installroot=/tmp/rpmtest basesystem
$ sudo yum --disablerepo=¥* --enablerepo=c6-media install --installroot=/tmp/rpmtest glibc
$ sudo yum --disablerepo=¥* --enablerepo=c6-media install --installroot=/tmp/rpmtest lvm2-libs
$ sudo rpm -ivh --root /tmp/rpmtest kpartx-0.4.9-46.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:kpartx                 ########################################### [100%]
$ ls /tmp/rpmtest/sbin/kpartx
/tmp/rpmtest/sbin/kpartx
$ rpm -qa | egrep "device-mapper-multi|kpartx"
kpartx-0.4.9-46.el6.x86_64
device-mapper-multipath-libs-0.4.9-46.el6.x86_64
device-mapper-multipath-0.4.9-46.el6.x86_64
$ rpm -qa --root /tmp/rpmtest | egrep "device-mapper-multi|kpartx"
kpartx-0.4.9-46.el6.x86_64
$
```

参考情報

- ソースパッケージのリポジトリサイト

  http://vault.centos.org/
- パッケージの検索:

  http://rpm.pbone.net/index.php3

  http://www.rpmfind.net/
- CentOS の HowTo に関するサイト:

  http://wiki.centos.org/HowTos/
- RPM に関するドキュメント

  http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/index.html

  http://docs.fedoraproject.org/en-US/Fedora/14/html/Software_Management_Guide/index.html