

# ***Software Packaging with RPM***

***Schalk W. Cronjé***  
***ysb33r@gmail.com***

# *Background*

- RPM == Red Hat Package Manager
- No longer just Red Hat
- Leadership by Marc Ewing and Erik Troan
- Development was in decay by start of decade
  - Back on track by combined efforts of main Linux vendors
- Not only for packaging traditional binary artefacts
  - also documents, java packages, perl modules etc.

# *Original Design Goals*

- Ease of use
- Package-oriented focus
- Upgradability of packages
- Tracking of package interdependencies
- Query capabilities
- Verification
- Support for multiple architectures
- Use of pristine sources

Source: RPM Guide <http://docs.fedoraproject.org>

# *RPM Contents*

- Compressed archive file
  - cpio format
- Installation instructions
  - Permissions and ownership to be applied to each file
  - Helper scripts (scriptlets)

# *RPM's place in the world*

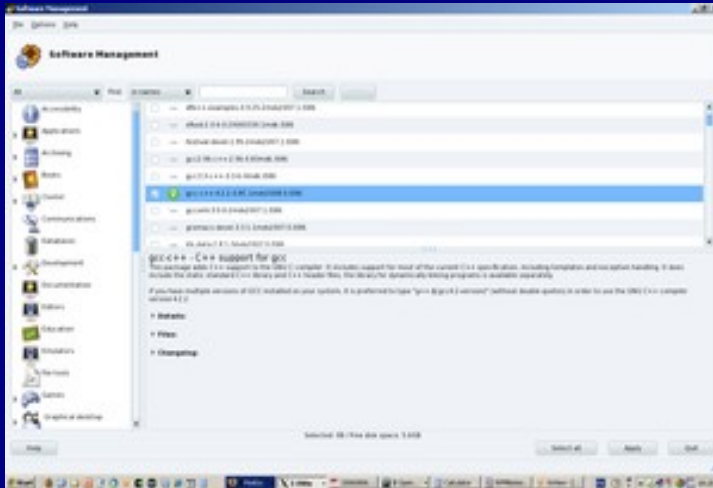
- Primary software package management for
  - RedHat
  - Suse
  - Mandriva
  - Fedora
  - CentOs
- Available on other platforms including:
  - Solaris
  - NetBSD
  - FreeBSD

# *Supported H/W Architectures*

- Intel compatible 32-bit (i386 ...)
- Intel compatible 64-bit (x86\_64)
- Intel Itanium (ia64)
- HP Alpha / Digital (alpha ...)
- Sparc/Ultra Sparc (sparc)
- ARM (armv3l ..)
- MIPS (mips, mipsel)
- Power PC (ppc ...)
- Motorola 68000 series (m68k ...)
- SGI MIPS (sgi)
- IBM RS6000 (rs6000)
- IBM S/390 (i370, s390x, s390 )

# RPM World

- RPM is not ...
  - yum
  - urpmi
  - rpmdrake



Repository Management Tools  
(urpmi, rpmdrake, yum ...)

perl

python

Base commands  
(rpm, rpmbuild ..)

RPM libraries

rpmdb  
(db4)

# *RPM's Peers*

- apt
  - Debian, Ubuntu
- portage
  - Gentoo
- pkgsrc
  - NetBSD, DragonBSD
- ports
  - FreeBSD



# Naming convention

```
# RPM macro convention  
%{name}-%{version}-%{release}.%{arch}.rpm
```

```
# Typical  
bc-1.06-20.i586.rpm
```

```
# Decorated with vendor distribution  
bc-1.06-20mdv2007.1.i586.rpm
```



**Vendor release**

```
# Source Package  
bc-1.06-20mdv2007.1.src.rpm
```

```
# Architecture independent binary package  
fonts-ttf-dejavu-2.15-2mdv2007.1.noarch.rpm
```

# ***Command-line Interface***

## ***Installing & Removal***

# Installing & Upgrading Packages

```
# Installation (requires root)
rpm -ivh bc-1.06-20mdv2007.1.i586.rpm
```

```
# Upgrading (requires root)
rpm -Uvh bc-1.06-20mdv2007.1.i586.rpm
```

**-U works for installation  
as well**

**Use -vh to produce visual  
feedback**

```
[#] rpm -Uvh bc-1.06-20mdv2007.1.i586.rpm
Preparing... ##### [100%]
```

# Removing Packages

```
# Removal (requires root)  
rpm -e NAME  
rpm --erase NAME
```

Only the name is required



```
[#] rpm -e bc
```

# Downgrading Packages

```
# Downgrading to older version (requires root)
rpm -Uvh --oldpackage NAME
```

```
[#] rpm -Uvh --oldpackage bc-1.05-20mdv2007.1.i586.rpm
Preparing... ##### [100%]
```

# ***Command-line Interface***

## ***Querying the database***

# *Is this Package Installed?*

```
# Is this package installed?  
rpm -q NAME
```

```
# Is this version of the package installed?  
rpm -q NAME-VERSION
```

```
[#] rpm -q bc  
bc-1.06-20mdv2007.1  
  
[#] rpm -q bcb  
package bcb is not installed  
  
[#] rpm -q bc-1.06  
bc-1.06-20mdv2007.1  
  
[#] rpm -q bc-1.07  
package bc-1.07 is not installed
```

# Getting a list of all packages

```
# Getting the list  
rpm -qa
```

```
[#] rpm -qa  
glibc-2.4-8mdv2007.1  
libcrack2-2.8.9-2mdv2007.1  
libattr1-2.4.32-2mdv2007.1  
libsqlite3_0-3.3.8-1mdv2007.1  
gawk-3.1.5-3mdv2007.1  
libkrb53-1.5.2-6.2mdv2007.1  
rpm-4.4.6-21mdv2007.1  
libx11_6-1.1.1-2.1mdv2007.1  
libfreetype6-2.3.1-3.1mdv2007.1  
libxrender1-0.9.2-1mdv2007.1
```



# Searching for Installed Packages

```
# Can use grep ...  
rpm -qa | grep GREP-PATTERN  
rpm -qa --pipe "grep GREP-PATTERN"  
  
# Or use internal pattern match  
rpm -qa 'GLOB'
```

```
[#] rpm -qa 'gcc*'  
gcc-cpp-4.2.2-0.RC.1mdv2008.0  
gcc-4.2.2-0.RC.1mdv2008.0  
gcc-c++-4.2.2-0.RC.1mdv2008.0
```

# Modifying display format

```
# Use --qf and RPM Macro Patterns
# to customise output
rpm -q --qf 'RPM-MACRO-PATTERN'
rpm -qa --qf '%{name}\n'
```

```
# Display the summary line
rpm -q --qf '%{summary}' NAME
```

```
[#] rpm -qa --qf '%{name}' 'gcc*'
gcc-cpp gcc gcc-c++
```

```
[#] rpm -qa --qf '%{name}: %{summary}\n' 'gcc*'
gcc-cpp: The C Preprocessor
gcc: GNU Compiler Collection
gcc-c++: C++ support for gcc
```

# Who Supplied this File?

```
# Use -f to refer to any installed filename  
rpm -qf FILENAME
```

```
[#] rpm -qf /usr/bin/zcat  
gzip-1.3.11-5mdv2007.1
```

```
[#] rpm -qf /home/schalk/accu.txt  
file /home/schalk/accu.txt is not owned by any package
```

# List of Files in Package

```
# Use -l  
rpm -q -l NAME
```

```
[#] rpm -ql bc  
/usr/bin/bc  
/usr/bin/dc  
/usr/share/doc/bc-1.06  
/usr/share/doc/bc-1.06/AUTHORS  
/usr/share/doc/bc-1.06/COPYING  
/usr/share/doc/bc-1.06/COPYING.LIB  
/usr/share/doc/bc-1.06/FAQ  
/usr/share/doc/bc-1.06/NEWS  
/usr/share/doc/bc-1.06/README  
/usr/share/info/bc.info.bz2  
/usr/share/info/dc.info.bz2  
/usr/share/man/man1/bc.1.bz2  
/usr/share/man/man1/dc.1.bz2
```

# Checking Installed Package Requirements

```
# --requires lists requirements of package  
rpm -q --requires NAME
```

Requires exact version  
of package

```
[#] rpm -q --requires gcc  
binutils >= 2.16.91.0.7-6mdk  
gcc-cpp = 4.2.2-0.RC.1mdv2008.0  
libgcc >= 3.3.2-5mdk  
update-alternatives  
glibc-devel >= 2.4-6mdk  
rpmLib(VersionedDependencies) <= 3.0.3-1  
libc.so.6  
libc.so.6(GLIBC_2.0)  
libc.so.6(GLIBC_2.1)  
libc.so.6(GLIBC_2.2)  
libc.so.6(GLIBC_2.3)  
libc.so.6(GLIBC_2.3.4)  
libc.so.6(GLIBC_2.4)  
rtld(GNU_HASH)
```

Requires minimum  
version of package

Needs library installed

# *What Does this Package Provide?*

```
# --provides lists symbolic names that  
# a package provides  
rpm -q --provides NAME
```

```
[#] rpm -q --provides gcc  
gcc4.2 = 4.2.2-0.RC.1mdv2008.0  
devel(libgcc_s)  
gcc = 4.2.2-0.RC.1mdv2008.0
```

# *Finding a Package Providing a Service*

```
# --whatprovides shows the package which provides  
# a specific symbolic name  
rpm -q --whatprovides SYMBOLIC_NAME
```

```
[#] rpm -q qt  
package qt is not installed  
  
[#] rpm -q --whatprovides qt  
qt3-common-3.3.8-4mdv2007.1
```

# Displaying Package Information

```
# --info dumps information about a package
# Usually Header + %description
rpm -q --info NAME
```

```
[#] rpm -q --info gcc-c++
Name       : gcc-c++                Relocations: (not relocatable)
Version    : 4.2.2                 Vendor: Mandriva
Release    : 0.RC.1mdv2008.0      Build Date: Sa 15 Sep 2007 02:56:09 BST
Install Date: Do 27 Mrt 2008 21:19:45 GMT Build Host: n4.mandriva.com
Group      : Development/C++      Source RPM: gcc-4.2.2-0.RC.1mdv2008.0.src.rpm
Size       : 8732938              License: GPLv3+
Signature  : DSA/SHA1, vr 05 okt 2007 00:56:31 BST, Key ID e7898ae070771ff3
Packager   : Pixel <pixel@mandriva.com>
URL        : http://gcc.gnu.org/
Summary    : C++ support for gcc
Description:
This package adds C++ support to the GNU C compiler. It includes support
for most of the current C++ specification, including templates and
exception handling. It does include the static standard C++
library and C++ header files; the library for dynamically linking
programs is available separately.
```



# Inspecting Uninstalled Package

```
# Same as for other queries
# Just add -p
rpm -qp -l
rpm -qp --provides
rpm -qp --requires
rpm -qp --info
```

```
[#] rpm -qpl /home/schalkc/yanything-1.1-3.i586.rpm
/etc/rc.d/init.d/yanything
/etc/yanything.conf
/usr/sbin/yanything
/usr/share/doc/yanything-1.1
/usr/share/doc/yanything-1.1/CHANGELOG
/usr/share/doc/yanything-1.1/COPYING
/usr/share/doc/yanything-1.1/README
/var/lib/yanything/ldap.group.pl
/var/lib/yanything/ldap.passwd.pl
```

# *Building RPMs*

# Rules

- **Don't build as *root*.**
- Setup a build environment
- **Don't build as *root*.**
- Ensure own spec files don't require *root* to build
- **Don't build as *root*.**
- Use macro names for system applications

# *Creating a Build Environment*

- Create .rpmrc
- Create .rpmmacros
- Create directories

# *.rpmrc*

```
buildarchtranslate: i386: i586  
buildarchtranslate: i486: i586  
buildarchtranslate: i586: i586  
buildarchtranslate: i686: i586
```

# ***.rpmmacros***

```
%_topdir           /home/schalkc/RPM
%_builddir         %{_topdir}/BUILD
%_tmppath          %{_topdir}/tmp
%_rpmdir           %{_topdir}/RPMS
%packager          Schalk <ysb33r@gmail.com>

# Add if building for a commercial company
%vendor            SomeCompany

# Add if signing RPMs
%_signature        gpg
%_gpg_path         ~/.gnupg
```

# Create Directories

# Create directory for delivering binary RPMs

mkdir -p /home/schalkc/RPM/RPMS ← **%{\_rpmmdir}**

# Create directory for delivering source RPMs

mkdir -p /home/schalkc/RPM/SRPMS ← **%{\_srcrpmdir}**

# Create directories used for building packages

mkdir -p /home/schalkc/RPM/BUILD ← **%{\_builddir}**

mkdir -p /home/schalkc/RPM/tmp ← **%{\_tmppath}**

# Create directory for source locations

mkdir -p /home/schalkc/RPM/SOURCES ← **%{\_sourcedir}**

# Create directory for spec files

mkdir -p /home/schalkc/RPM/SPECS ← **%{\_specdir}**

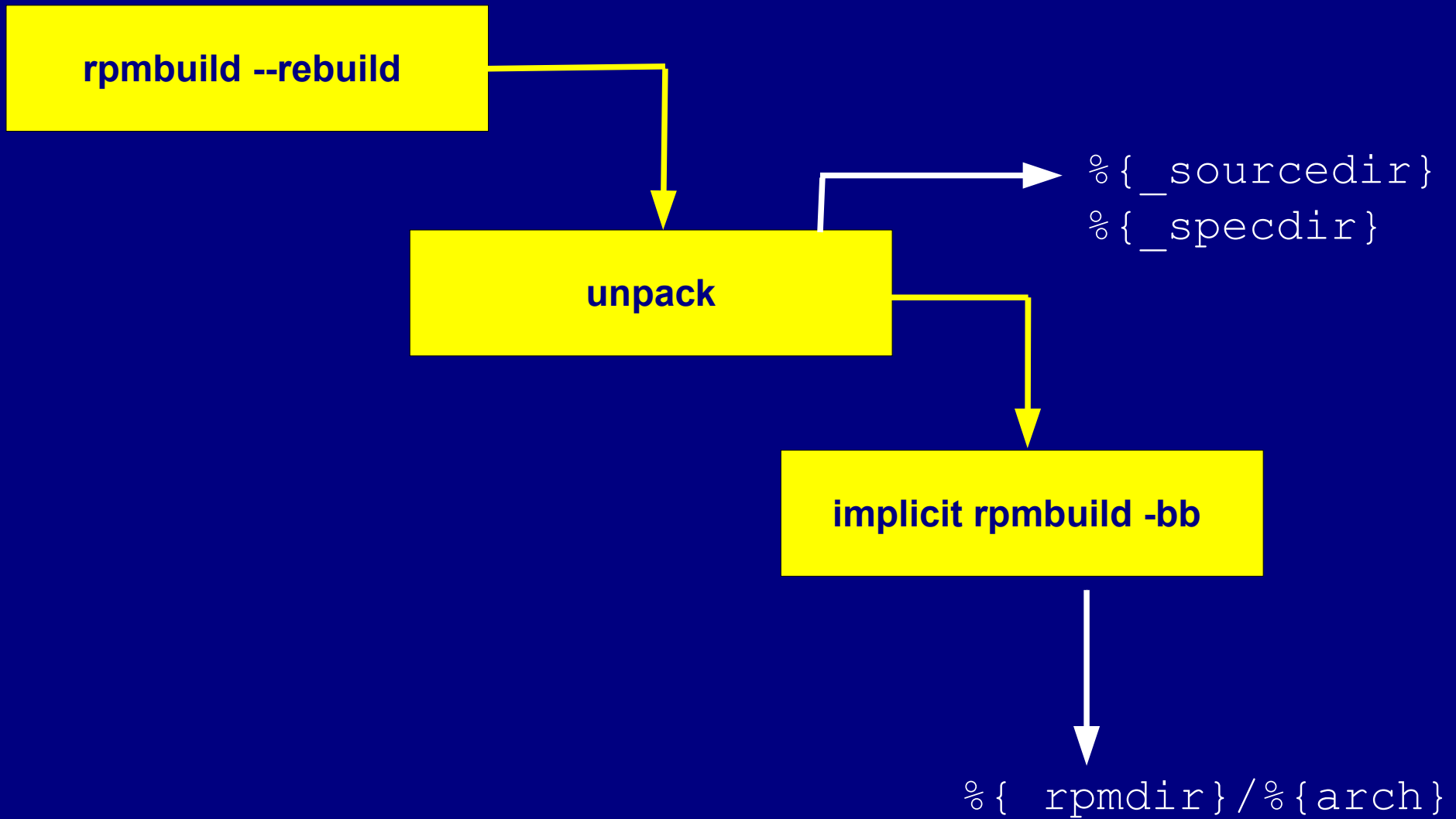
# *Rebuilding a Source Package*

- Motivation
  - No binary package available
  - Existing binary package will not install
  - Customisation via command-line

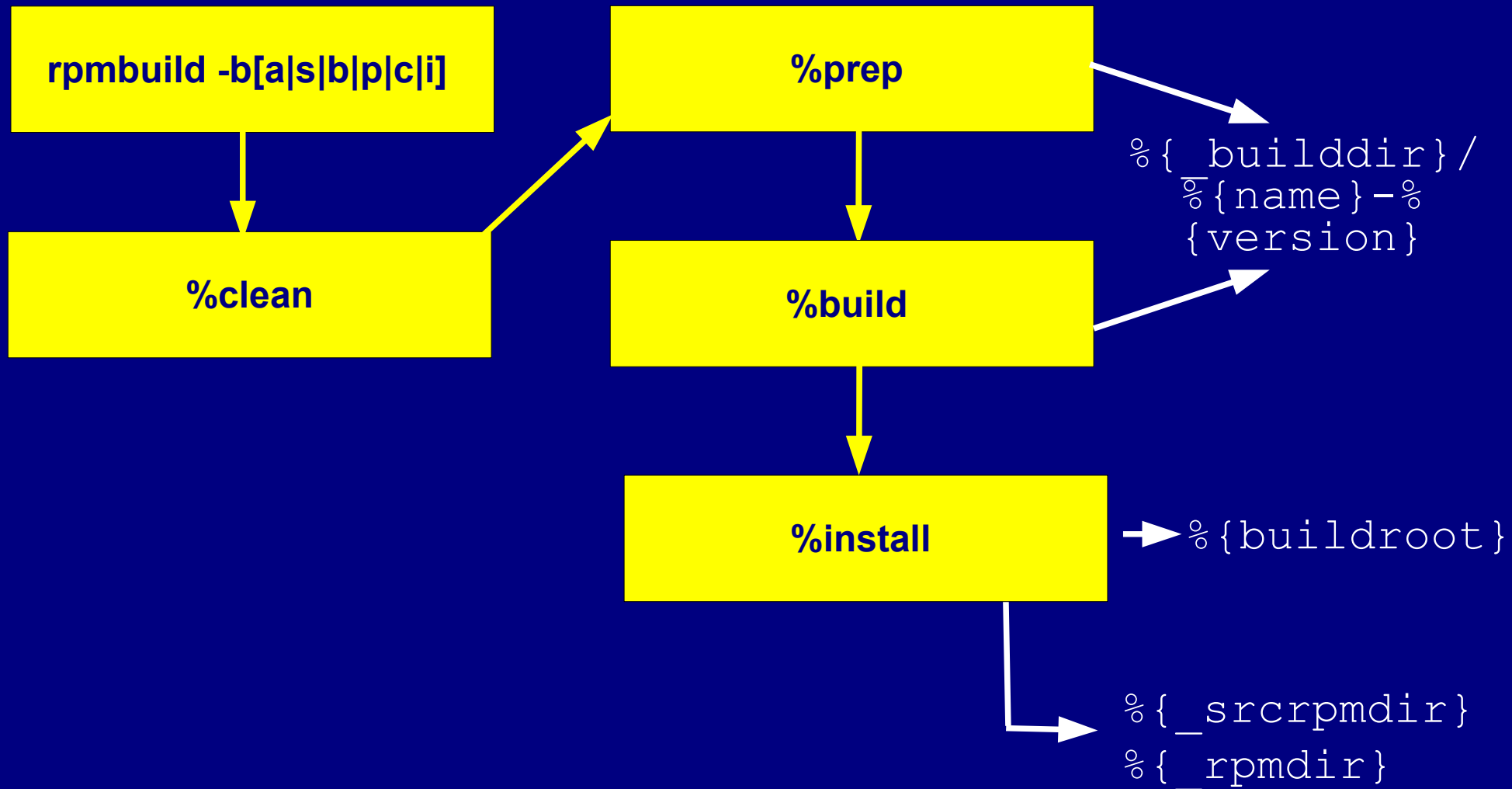
```
# Use --rebuild command  
rpmbuild --rebuild bc-1.06-20mdv2007.1.src.rpm
```



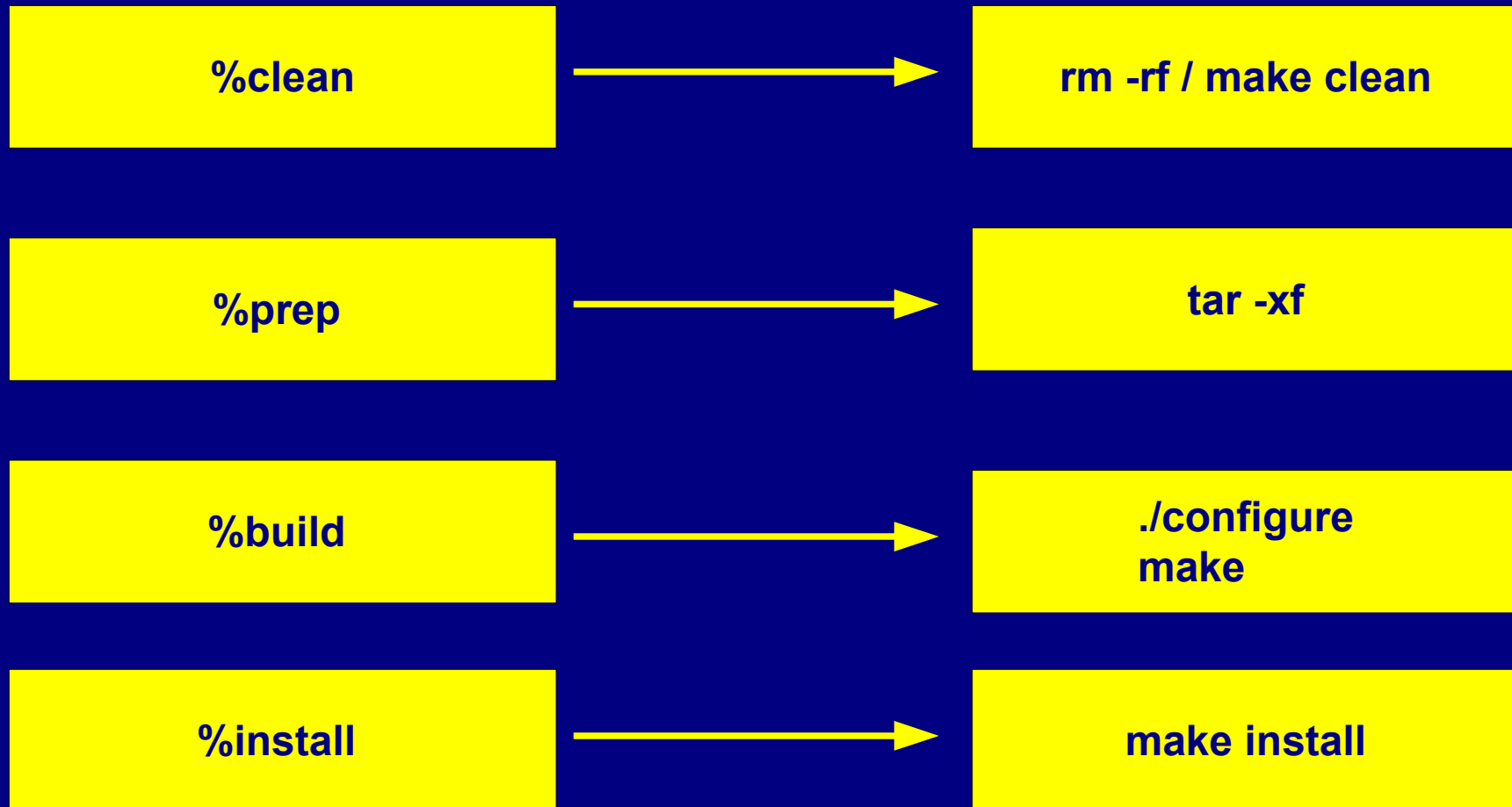
# Source Rebuild Process



# Build Process



# Mapping RPM to OSS Convention



# *Building Packages*

```
# Build everything  
rpmbuild-ba /path/to/specfile
```

```
# Build binary RPMs  
rpmbuild -bb /path/to/specfile
```

```
# Build source RPM only  
rpmbuild -bs /path/to/specfile
```

# Intermediate Build Commands

```
# Setup, but don't build (Invokes %prep)
rpmbuild -bp /path/to/specfile
```

```
# Setup and build (Invokes %build)
rpmbuild -bc /path/to/specfile
```

```
# Setup, build and install, but don't package
rpmbuild -bi /path/to/specfile
```

```
# To skip a previous stage
rpmbuild --short-circuit -b[p|c|i] /path/to/spec-file
```

# *Spec Files*

# Variables & Macros

- Defined using `define variable value`
- Used as `%{variable}`
- If not found expands to literal string
- `%% => %`
- System macros start with `_` or `__`

```
define myvar myvalue
# will print 'myvalue'
echo %{myvar}

# will print '%{myvar}'
echo %{myvar}

# will print '%{myvar}'
echo %%{myvar}
```

# *Conditional Sections*

- Conditionals were limited in older versions
- Later versions have added special conditionals:
  - %ifos
  - %ifarch



# *Equivalent of #if*

```
# The old way to do the equivalent of #if
```

```
%if %{?my_variable:1}%{!?my_variable:0}
```

```
...
```

```
%else
```

```
...
```

```
%endif
```

```
# The new way to do the equivalent of #if
```

```
%if %{my_variable}
```

```
...
```

```
%else
```

```
...
```

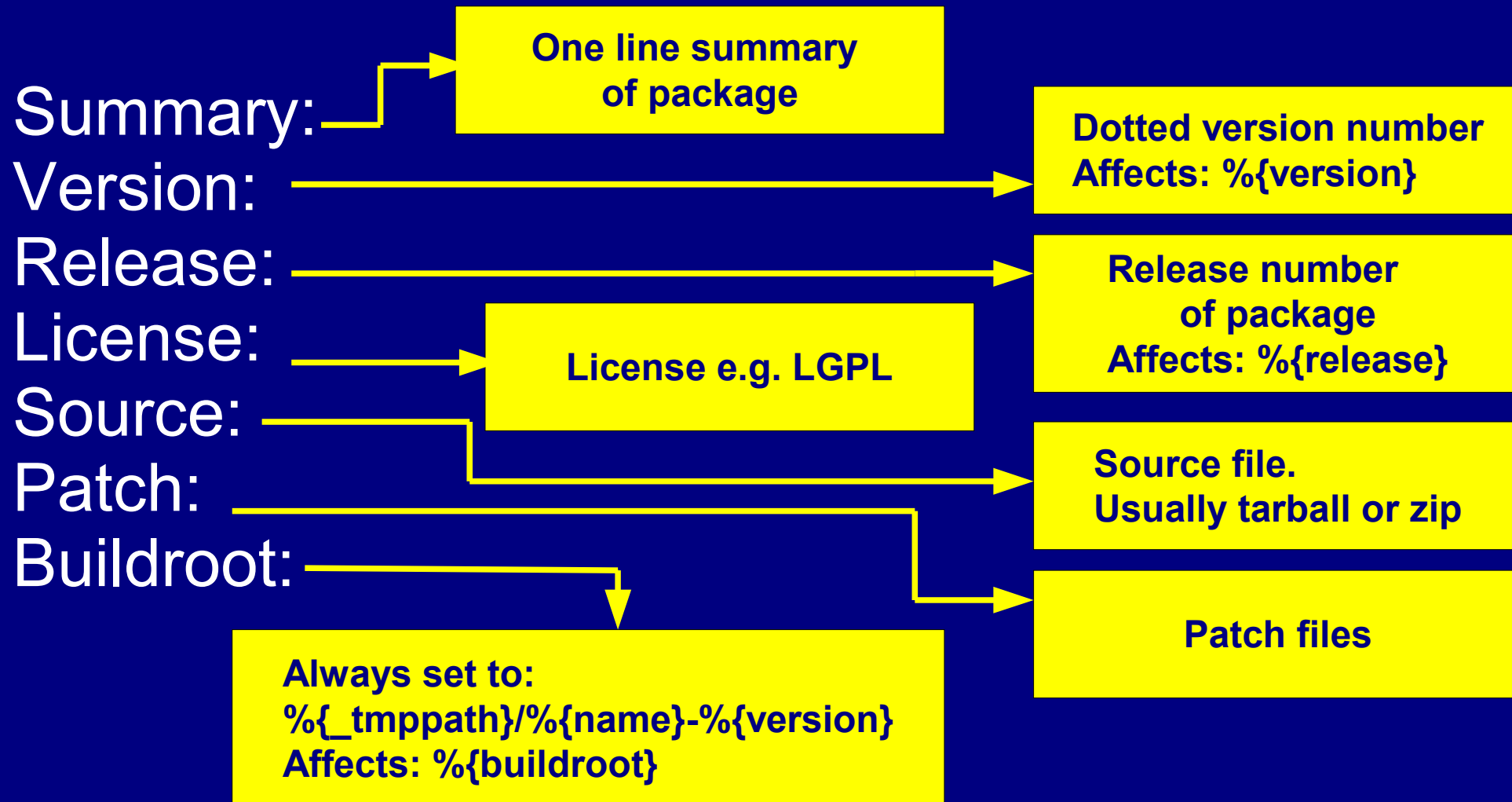
```
%endif
```

# *Spec File Layout*

- Header
- Description
  - %description
- Build
  - %prep, %build, %install, %clean
- Package:
  - %files, %verify, scriptlets
- Changelog
  - %changelog

# Spec File Header

A number of fields in the format `name: value`



**Summary:** GNU's bc (a numeric processing language)  
**Name:** bc  
**Version:** 1.06  
**Release:** 6  
**License:** GPL  
**URL:** <http://www.gnu.org/software/bc/bc.html>  
**Group:** Sciences/Mathematics  
**Source:** <ftp://ftp.gnu.org/gnu/bc/bc-#{version}.tar.bz2>  
**Patch0:** bc-1.06-readline42.patch.bz2  
**BuildRequires:** flex gpm-devel ncurses-devel readline-devel  
**Buildroot:** %{\_tmppath}/%{name}-#{version}

# Description

- To provide a multi-line description of a package
- Usually taken from README

```
%description
```

```
The bc package includes bc and dc. Bc is an arbitrary precision numeric processing arithmetic language. Dc is an interactive arbitrary precision stack based calculator, which can be used as a text mode calculator.
```

```
Install the bc package if you need its number handling capabilities or if you would like to use its text mode calculator.
```

# *Preparation Phase (%prep)*

- Unpacks & decompresses all sources via `%setup`
- Assumes that destination directory is `%{_builddir}/%{name}-%{version}`
- Decompresses and applies all patches via `%patch`
- `%setup / %setup0 => Source, Source0`
- `%setup1 => Source1 ...`
- `%patch / %patch0 => Patch, Patch0`
- `%patch1 => Patch1 ...`

# *Example %prep*

```
%prep  
%setup -q  
%patch -p1 -b .readline42
```

# ***Build Phase (%build)***

- Performs the tasks associated with configuring and building
- Most distributions provide a special %make macro to help with Autoconf environments
- Direct access to GNU Make via %{\\_make}



# ***Build Installation Phase (%install)***

- Performs the actions of installing software into a special root
- Software is packaged up from this root according to %files
- Most distributions provide a %makeinstall macro to help with AutoConf environments

# *Example %build, %install, %clean*

```
%build  
%configure --with-readline  
%make LDFLAGS=-s
```

```
%install  
rm -rf %{builddroot}  
%makeinstall
```

```
%clean  
rm -rf %{builddroot}
```

# *File Management (%files)*

- Specifies which files have to be packaged up
- Will error if:
  - File specified, but not found
  - File found, but not specified
- Can specify default modes & ownership
  - Use `%defattr`
- Can specify mode & ownership per file
  - Use inline `%attr`

# Example %files

```
%files
%defattr(-,root,root)
%doc COPYING COPYING.LIB FAQ AUTHOR NEWS README
%{_bindir}/bc
%{_bindir}/dc
%{_mandir}/man1/bc.1*
%{_mandir}/man1/dc.1*
%{_infodir}/dc.info*
```

# Overriding Attributes

```
## specific permissions and ownerships can be changed from default
## by using %attr
%files
%defattr(-,root,root)
%attr(0711,nullmail,nullmail) %[_sbindir]/nullmailer-queue
```

# *Earmarking Configuration Files*

```
## Mark configuration files as %config
## Prevents --verify from reporting MD5 errors.
```

```
%files
%config %{_sysconfdir}/ldap.conf
```

```
## Add 'noreplace' to prevent upgrades from overwriting a config file
## Upgrade will create a file with a .rpmsave suffix
```

```
%files
%config(noreplace) %{_sysconfdir}/ldap.conf
```

```
## Can even be combined with %attr
```

```
%files
%config(noreplace) %attr(0640,root,ldap) %{_sysconfdir}/ldap.conf
```

# *Ignoring Installed Files*

```
## Some installed files need not be packaged
## If not explicitly called out installation phase will fail

%files
%exclude %{_sysconfdir}/ldap.conf.example
```

# *Packaging Empty Directory*

```
## By default rpmbuild will not package empty directories
## To package an empty directory use %dir
%files
%defattr (-,root,root)
%dir %[_locatestatedir]/yanything

## Specific permissions and ownership is possible
%files
%defattr (-,root,root)
%dir %attr(0700,daemon,daemon) %[_locatestatedir]/yanything
```



# *Pre-installation scriptlet (%pre)*

- Allows for work to be done before package transaction is started
- If scriptlet exits non-zero installation is aborted
- Can use count to determine whether installation or upgrade
- Shell type is 'sh' not 'bash'

# *%pre*

```
## Can use conditional:  
## 1 is first-install,  
## >1 is upgrade  
%pre  
echo 'This is entering pre-package installation'  
  
if [ "$1" = 1 ] ; then  
    echo 'This is the first time this package is installed'  
fi
```

# *Post-installation scriptlet (%post)*

- Is executed after all files have installed on system
- Can call installed files
- Useful for updating SYSV initscripts.
- If scriptlet exists zero, warning is issued, but installation continues.
- Can use count to determine whether installation or upgrade
- Shell type is 'sh' not 'bash'

# *%post*

```
## Can use conditional:  
## 1 is first-install,  
## >1 is upgrade  
%post  
echo 'This is entering post installation phase'  
  
if [ "$1" = 1 ] ; then  
    /sbin/chkconfig --add nullmailer  
    /sbin/chkconfig nullmailer on  
else  
    /sbin/service nullmailer condrestart  
fi
```

# *Pre-uninstall scriptlet (%preun)*

- Is executed before a package is uninstalled
- Can call installed files
- Useful for stopping services.
- If scriptlet exists zero, operation is aborted.
- Can use count to determine whether installation or upgrade
- Shell type is 'sh' not 'bash'

# *%preun*

```
## Can use conditional:  
## 0 is final removal,  
## >0 is downgrade  
%preun  
echo 'This is entering pre-uninstall phase'  
  
if [ "$1" = 0 ]; then  
    /sbin/chkconfig nullmailer off  
    /sbin/chkconfig --del nullmailer  
fi
```

# *Post-uninstall scriptlet (%postun)*

- Is executed after all files have been removed
- Useful for final cleanup
- If scriptlet exists zero, warning is printed, but operation succeeds
- Can use count to determine whether installation or upgrade
- Shell type is 'sh' not 'bash'

# *%postun*

```
## Can use conditional:  
## 0 is final removal,  
## >0 is downgrade  
%postun  
echo 'This is entering post-uninstall phase'  
  
if [ "$1" = 0 ]; then  
    %{_sbindir}/userdel nullmail  
    %{_sbindir}/groupdel nullmail 2>/dev/null || /bin/true  
fi
```



# *Multiple Packages*

- Multiple packages allows flexibility to build once, but install selectively
- Few other packaging systems have this feature
- Additional packages are indicated by using `%package`
- Use `-n` in order to change stem
- Each package requires a `%description`
- Each package may have custom `%files` and scriptlets

```
## This spec will produce:  
## nullmailer-1.00-2.i386.rpm  
## nullmailer-sendmail-wrapper-1.00-2.i386.rpm  
## nullmailer-mailq-wrapper-1.00-2.i386.rpm
```

```
Name: nullmailer
```

```
Summary: Simple relay-only mail transport agent
```

```
Version: 1.00
```

```
Release: 2
```

```
...
```

```
BuildRoot: %{_tmppath}/%{name}-%{version}
```

```
Provides: smtpdaemon smtp-daemon
```

```
Conflicts: sendmail
```

```
Conflicts: qmail
```

```
%package sendmail-wrapper
```

```
Summary: Sendmail wrapper for nullmailer.
```

```
Group: Networking/Mail
```

```
Conflicts: courier-sendmail-wrapper
```

```
Requires: %{name} = %{version}
```

```
%package mailq-wrapper
```

```
Summary: Mailq wrapper for nullmailer.
```

```
Group: Networking/Mail
```

```
Requires: %{name} = %{version}
```

```
## within -n this spec will produce:  
## nullmailer-1.00-2.i386.rpm  
## sendmail-wrapper-1.00-2.i386.rpm  
## mailq-wrapper-1.00-2.i386.rpm
```

```
Name: nullmailer
```

```
Summary: Simple relay-only mail transport agent
```

```
Version: 1.00
```

```
Release: 2
```

```
...
```

```
BuildRoot: %{_tmppath}/%{name}-%{version}
```

```
Provides: smtpdaemon smtp-daemon
```

```
Conflicts: sendmail
```

```
Conflicts: qmail
```

```
%package -n sendmail-wrapper
```

```
Summary: Sendmail wrapper for nullmailer.
```

```
Group: Networking/Mail
```

```
Conflicts: courier-sendmail-wrapper
```

```
Requires: %{name} = %{version}
```

```
%package -n mailq-wrapper
```

```
Summary: Mailq wrapper for nullmailer.
```

```
Group: Networking/Mail
```

```
Requires: %{name} = %{version}
```

**## A custom %description is required per package**

**%description**

Nullmailer is a mail transport agent designed to only relay all its messages through a fixed set of "upstream" hosts. It is also designed to be secure.

**%description sendmail-wrapper**

Provides a sendmail wrapper for applications that looks for /usr/lib/sendmail or /usr/sbin/sendmail

**%description -n mailq-wrapper**

Provides a mailq wrapper so that information can be retrieved using the mailq command.

```
## %files and customisation can occur per package

## This applies to the base package
%files

## These are files for nullmailer-sendmail-wrapper
%files sendmail-wrapper

## These are files for mailq-wrapper (note -n)
%files -n mailq-wrapper

## This is %pre for the base package
%pre

## This is a %postun for mailq-wrapper (note -n)
%postun -n mailq-wrapper
```

# *CPAN Modules - The Quick Way*

- In RPM-based system CPAN Perl modules should be installed using RPM
- Provides for better dependency management
- Most CPAN modules can be built by customising a simple template

**NOTE:** Same approach applies for other language modules i.e. Python, Ruby.

```
## Set the name of the package as known in CPAN. Replace any :: with -
%define cpan_package Net-IPv4Addr

## Set the Version from CPAN
Version: 0.10

## Set the release number. This starts at 1, bump as needed
Release: 1

## A one-liner description
Summary: Perl extension for parsing IPv4 addresses.

Name: perl-%{cpan_package}
Group: Development/Perl
License: GPL or Artistic
Source: %{cpan_package}-%{version}.tar.gz
BuildRoot: %{_tmppath}/%{name}-%{version}-%{release}

%description
Net::IPv4Addr provides functions ...

## ... more to follow ...
```

```
%prep
%setup -q -n %{cpan_package}-${version}

%build
%{__perl} Makefile.PL INSTALLDIRS="vendor"
%{__make}

%install
%{__rm} -rf $RPM_BUILD_ROOT
%{__make} pure_install DESTDIR=%{buildroot}

%clean
%{__rm} -rf $RPM_BUILD_ROOT

%files
%defattr(-,root,root)
%doc README
%{_bindir}/ipv4calc
%{perl_vendorlib}/*
%{_mandir}/man*//*

%changelog
* Mon Jul 17 2007 Schaik w. Cronje 1.10-1
- Created spec file
```



# *Commercial Packages*

# *Commercial Requirements*

- Might not want to distribute source
- Need agreement to license before install
- Might not use spec-driven build
- Linux packaging as an after-thought

# *Source RPMs with Limited Source*

- It is possible to build SRPMs that do not contain all of the source
- Allows for distribution of all non-restricted components
- Restricted components can then be obtained under NDA / licence / payment
- Allows for multi-configuration distribution
- Header fields:
  - NoSource
  - NoPatch

# *Using NoPatch with Conditional*

```
Source0:          our-standard-code.tar.bz2
NoPatch0:         special-performance-patch-against-payment.patch.bz2

%prep
%setup -q
%if %{?with_performance_patch:1}%{!with_performance_patch:0}
%patch
%endif
```

# *Binary-only RPMs*

- It is always just possible to build binary RPMs without distributing SRPMs
- Integrate `rpmbuild -bb` as part of build process
- This is the recommended solution

# *Packaging up Files Only*

- Environment might produce artefacts via another build system
- Distribution via RPM is an afterthought
- Solution is to create a list of files via build system, then feed this to spec file
- Use %files -f

# Packaging up Files Only

```
Summary:      A scanner for HTTP
Name:         mycompany-http-scanner
Version:      1.06
Release:      6
License:      MyCompany's Commercial License
Group:        Networking/Daemon
Buildroot:    %{_tmppath}/%{name}-%{version}
```

```
%description
```

```
We are providing this incredible, best since sliced bread, HTTP scanner.
```

```
%prep
```

```
%build
```

```
%install
```

```
%files -f /dev/fd/0
```

```
[#] cat myfilelist.txt | rpmbuild -bb mycompany-http-scanner.spec
```

# Turning off AutoReqProv

- Due to packaging method, automatic *requires/ provides* might not be feasible
- Responsibility with packager to manually provide requirements
- Header fields:
  - AutoReqProv
  - Requires
  - Provides

```
AutoReqProv: no  
Requires: libgcc >= 3.3.2  
Provides: http-scanner
```



***More Tips & Tricks***  
***(not for the faint of heart)***

# *Installing Custom Macros*

- To help other build packages against your package install your own macros
- Create a file named the same as your top-level package
- Install into /etc/rpm.d
- Will be read during rpmbuild initialisation
- **Choose macro names carefully**
  - Remember: No namespace support
  - Approach naming similar to C macros

# Calling RPM from GNU Make

```
# Assume BUILDDIR was set in Makefile

RPMDIR=$(BUILDDIR)/rpm

rpm:
  mkdir -p $(RPMDIR)/rpm $(RPMDIR)/build \
    $(RPMDIR)/tmp && \
  MAKEFLAGS= \
  rpmbuild -bb \
    --define "_sourcedir $(CURDIR)" \
    --define "_rpmdir $(RPMDIR)/rpm" \
    --define "_builddir $(RPMDIR)/build" \
    --define "_tmppath $(RPMDIR)/tmp" \
    MyProjectSpec.spec
```

# Calling RPM from GNU Make #2

```
## Define GNU Make macro
rpmbuild=mkdir -p $2/rpm $2/build $2/tmp && \
  MAKEFLAGS= \
    rpmbuild -bb \
      --define "_sourcedir $1" \
      --define "_rpmdir $2/rpm" \
      --define "_builddir $2/build" \
      --define "_tmppath $2/tmp" \
      $3

RPMDIR=$(BUILDDIR)/rpm

rpm:
  @$(call rpmbuild,$(CURDIR),$(RPMDIR),MyProject.spec)
```

# Using SVN revision as RPM Release

```
[#] rpm -ba --define "m_release $(svn info 2>/dev/null | \
    grep ^Revision | awk '{print $2}')" MyProject.spec
```

# In GNU Make

```
RPMRELEASE=$(strip $(shell svn info 2>/dev/null | grep ^Revision | awk
    '{print $$2}'))
```

# SVN + Gmake + RPM (1 of 2)

```
rpmbuild=mkdir -p $2/rpm $2/build $2/tmp && \  
  MAKEFLAGS= \  
  rpmbuild -bb \  
    --define "_sourcedir $1" \  
    --define "_rpmdir $2/rpm" \  
    --define "_builddir $2/build" \  
    --define "_tmppath $2/tmp" \  
    $(if $4,--define "m_release $4") \  
    $3  
  
RPMRELEASE=$(strip $(shell svn info 2>/dev/null | grep ^Revision | awk  
'{print $$2}'))  
  
rpm:  
  $(call rpmbuild,$(CURDIR),$(RPMDIR),\  
    MyProject.spec,$(RPMRELEASE))
```

# SVN + Gmake + RPM (2 of 2)

```
## Set the version of this build.  
## It can be overridden from the Makefile  
## if necessary  
%if %{?m_release:1}%{!m_release:0}  
Release: %{m_release}  
%else  
Release: 1  
%endif
```

# Re-entering Gmake from RPM (1 of 2)

- Possibility of combining RPM build process into Gmake, but reusing same Makefile from within spec file
- Allows for less maintenance





# Re-entering Gmake from RPM (2 of 2)

```
## Spec-file
%define crondir %[_sysconfdir]/cron.d
%define apacheconfdir %[_sysconfdir]/httpd/conf.d
%define statedir %[_localstatedir]/lib/my-project
%define perldir %[_perl_privlib]
%define perlmodroot %[_perldir]/McAfee/B1etchley

%define BUILDSCRIPT_OPTS1 PREFIX_LOCALESTATEDIR=%[_statedir]
%define BUILDSCRIPT_OPTS2 PREFIX_PERLMODROOT=%[_perldir] PREFIX_CRONDIR=%
    {crondir}
%define BUILDSCRIPT %[___make] -C %[_sourcedir] %[_BUILDSCRIPT_OPTS1] %
    {BUILDSCRIPT_OPTS2} PREFIX_VERSION=%[_version]

%build
%[_BUILDSCRIPT] BUILDROOT=%[_builddir]

%install
%[___rm] -rf %[_buildroot]
%[_BUILDSCRIPT] install INSTALL_DIR=%[_buildroot]
```

# Querying Spec File for Artefacts

```
# Use rpmquery --specfile
#
rpmquery --qf '%{name}-%{version}-%{release}.%{arch}.rpm' --specfile SPECFILENAME
```

```
[#] rpmquery --qf '%{name}-%{version}-%{release}.%{arch}.rpm' --specfile \
      ypanything.spec
ypanything-1.1-3.i586.rpm

[#] rpmquery --qf '%{name}-%{version}-%{release}.%{arch}.rpm' --specfile \
      nullmailer.spec
nullmailer-1.00-0.RC7.2mdk.i586.rpm
nullmailer-sendmail-wrapper-1.00-0.RC7.2mdk.i586.rpm
nullmailer-mailq-wrapper-1.00-0.RC7.2mdk.i586.rpm
```

# Unpacking Contents

```
# Unpack content of RPM-contained archive  
rpm2cpio PACKAGE | cpio -id
```

```
[#] rpm -qpl ypanything-1.1-3.i586.rpm  
/etc/rc.d/init.d/ypanything  
/etc/ypanything.conf  
/usr/sbin/ypanything  
/usr/share/doc/ypanything-1.1  
/usr/share/doc/ypanything-1.1/CHANGELOG  
/usr/share/doc/ypanything-1.1/COPYING  
/usr/share/doc/ypanything-1.1/README  
/var/lib/ypanything/ldap.group.pl  
/var/lib/ypanything/ldap.passwd.pl  
  
[#] rpm2cpio ypanything-1.1-3.i586.rpm | cpio -id  
152 blocks  
  
[#] ls  
/etc /usr /var ypanything-1.1-3.i586.rpm
```

# *Don't Strip my Binaries!*

- RPM strips symbols by default
- Override this behaviour by redefining `__os_install_post`
- Similar behaviour customisation possible for other macros

```
%define __os_install_post %{nil}
```

# What are the Scriptlets?

```
# Query scriptlets of an installed package
rpm -q --scripts NAME
rpm -q -p --scripts PACKAGE
rpm -q --scripts --specfile SPECFILE
```

```
[#] rpm -q --scripts -p /ypanything-1.1-3.i586.rpm
preinstall program: /bin/sh
postinstall scriptlet (using /bin/sh):
if [ "$1" = 1 ]; then
    chkconfig --add ypanything
fi
preuninstall scriptlet (using /bin/sh):
if [ "$1" = 0 ]; then
    chkconfig ypanything off
    chkconfig --del ypanything
fi
postuninstall program: /bin/sh
```

no %pre

%post

%preun

no %postun

# *Working Around Failing Scriptlets*

```
# Any scriptlet or trigger scripts can be disabled
rpm -Uvh --noscripts PACKAGE
rpm -Uvh --notriggers PACKAGE
rpm -Uvh --nopre PACKAGE
rpm -Uvh --nopost PACKAGE
rpm -e --noscripts NAME
rpm -e --notriggers NAME
rpm -e --nopreun NAME
rpm -e --nopostun NAME
```

```
[#] rpm -e --nopreun yanything
```

# *Final Tips*

- See `/usr/lib/rpm/macros` for raw definitions of macros
- Use `rpm --showrc` to list macros
- Understand how to use `rpm -V` to validate files after installation and for simple security audits
- Always use `%{__command}` macros instead of commands.

***Thank you***

***May you successfully build RPMs!***