

# Собираем RPM пакет своими руками

## Вступление

В первой части статьи вы ознакомились с [основами работы RPM](#). Но часто бывает так, что вам необходимо собрать пакет с необходимыми опциями (включить поддержку mysql, postgresql или cyrus-sasl2 и т.п.), которые отсутствуют в rpm пакете, поставляемом на диске с дистрибутивом. Выходом из этой ситуации является сборка своего собственного пакета.

Для облегчения сборки rpm пакетов существует специально предназначенный для этих целей пакет - rpm-build.

```
# rpm -qi rpm-build
Name           : rpm-build           Relocations: (not relocatable)
Version        : 4.3.3              Vendor: CentOS
Release        : 7_nonptl           Build Date: Пнд 21 Фев 2005 20:21:52
Install Date:  Сбт 09 Апр 2005 22:14:57  Build Host: guru.build.karan.org
Group          : Development/Tools    Source RPM:
rpm-4.3.3-7_nonptl.src.rpm
Size           : 1576124             License: GPL
Signature      : DSA/SHA1, Вск 27 Фев 2005 00:36:59, Key ID a53d0bab443e1821
Packager       : Karanbir Singh
Summary        : Scripts and executable programs used to build packages.
Description    :
The rpm-build package contains the scripts and executable programs
that are used to build packages using the RPM Package Manager.
```

Как видно из описания этот пакет содержит набор скриптов и программ, предназначенных для сборки пакетов.

Для того, чтобы собрать какой-либо пакет для начала необходимо загрузить т.н. исходники для сборки пакета, как правило, это файлы с расширением src.rpm. Иногда, как в случае с courier-imap, spec файл включается в исходные коды.

Очень удобным для поиска rpm и src.rpm пакетов является сайт [www.rpmfind.net](http://www.rpmfind.net). Например, мы нашли необходимый нам пакет - postfix, squid и т.д. Мы сразу можем узнать какие пакеты, необходимы для его сборки. Вот стандартная страница с информацией о пакете для postfix и для squid. Также там указывается контрольная сумма для проверки целостности пакета.

После того, как мы получили исходники и проверили их целостность, необходимо установить соответствующий пакет.

```
# rpm -ivh postfix-2.2.8-1.2.src.rpm
 1:postfix          #####
[100%]
```

После выполнения данной операции исходники postfix и все необходимые пачти, а также скрипты были установлены в [/usr/src/redhat/SOURCES/](#), а spec файл (инструкция для сборки

rpm пакета) в [/usr/src/redhat/SPECS/](#).

```
# ls /usr/src/redhat/SOURCES/  
pflogsumm-1.1.0.tar.gz      postfix-etc-init.d-postfix  
postfix-2.1.1-config.patch postfix-hostname-fqdn.patch  
postfix-2.1.1-obsolete.patch postfix-large-fs.patch  
postfix-2.1.5-aliases.patch postfix-pam.conf  
postfix-2.2.5-cyrus.patch  postfix-sasl.conf  
postfix-2.2.8.tar.gz      README-Postfix-SASL-RedHat.txt  
postfix-alternatives.patch  
  
# ls /usr/src/redhat/SPECS/  
postfix.spec
```

Это стандартное месторасположение файлов при установке src.rpm. В принципе названия папок говорят сами за себя. Структура папок

- [/usr/src/redhat/BUILD](#) - В данную папку распаковываются исходные коды соответствующего пакета. По умолчанию после сборки пакета исходные коды не удаляются, для того, чтобы их удалить, необходимо при сборке указать ключ **-clean**.
- [/usr/src/redhat/RPMS/arch-name](#) - где arch-name имя архитектуры и может принимать одно из следующих значений: **athlon, i386, i486, i586, i686, noarch**. После завершения сборки готовый rpm пакет будет помещен в соответствующую папку, в зависимости от того, какую вы зададите архитектуру при сборке пакета с помощью ключа **-target**.
- [/usr/src/redhat/SOURCES](#) - после установки src.rpm пакета в данную папку помещаются непосредственно исходные коды самого пакета, а также различные патчи и скрипты. Если вам надо наложить какой то патч, то его необходимо копировать именно в данную папку.
- [/usr/src/redhat/SPECS](#) - в этой папке располагается т.н. spec файл, в котором указывается последовательность действий для сборки rpm пакета.
- [/usr/src/redhat/SRPMS](#) - после окончания сборки rpm пакета в данную папку будет записан соответствующий src.rpm пакет.

И так, для того, чтобы начать собирать пакет необходимо перейти в папку с spec файлом и выполнить следующую команду

```
# cd /usr/src/redhat/SPECS/  
# rpmbuild -ba --target=i686 postfix.spec  
Платформы для сборки: i686  
Сборка для платформы i686  
Выполняется(%prep): /bin/sh -e /var/tmp/rpm-tmp.82019  
+ umask 022  
+ cd /usr/src/redhat/BUILD  
+ umask 022  
+ cd /usr/src/redhat/BUILD  
+ rm -rf postfix-2.2.8  
+ /bin/gzip -dc /usr/src/redhat/SOURCES/postfix-2.2.8.tar.gz  
+ tar -xf -  
+ STATUS=0
```

```
+ '[' 0 -ne 0 ']'
+ cd postfix-2.2.8
++ /usr/bin/id -u
+ '[' 0 = 0 ']'
+ /bin/chown -Rhf root .
++ /usr/bin/id -u
+ '[' 0 = 0 ']'
+ /bin/chgrp -Rhf root .
+ /bin/chmod -Rf a+rX,u+w,g-w,o-w .
+ echo 'Patch #1 (postfix-2.1.1-config.patch):'
Patch #1 (postfix-2.1.1-config.patch):
+ patch -p1 -b --suffix .config -s
...
...
...
Записан: /usr/src/redhat/SRPMS/postfix-2.2.8-1.2.src.rpm
Записан: /usr/src/redhat/RPMS/i686/postfix-2.2.8-1.2.i686.rpm
Записан: /usr/src/redhat/RPMS/i686/postfix-pflogsumm-2.2.8-1.2.i686.rpm
Выполняется(%clean): /bin/sh -e /var/tmp/rpm-tmp.73987
+ umask 022
+ cd /usr/src/redhat/BUILD
+ cd postfix-2.2.8
+ /bin/rm -rf /var/tmp/postfix-buildroot
+ exit 0
```

Из последних строчек видно, что готовый rpm пакет называется postfix-2.2.8-1.2.i686.rpm и сохранен в папке **/usr/src/redhat/RPMS/i686/**, так как при сборке пакета мы указали ключ **-target=i686**.

Собственно сборка не должна вызвать никаких проблем. Но что если нам необходимо собрать пакет со своими опциями, например, включить поддержку mysql или sasl2 и т.п.? Для этих целей необходимо будет подправить спес файл.

## Структура спес файла

Структурно спес файл состоит из следующих разделов.

### Заголовок

- **Summary:** - однострочное описание пакета.
- **Name:** - строка имени из имени файла rpm
- **Version:** - номер версии пакета
- **Release:** - номер релиза пакета
- **Icon:** - имя файла иконки, которое будет использоваться другими высокоуровневыми утилитами установки (подобными glint)
- **Source:** - указывает место расположения файлов исходных текстов.
- **Patch:** - Это место где вы можете найти заплатки, если вы захотите загрузить их снова.
- **Copyright:** - авторские права (GPL, BSD, MIT, public domain, distributable, или commercial)
- **BuildRoot:** - задает корневую директорию для построения и установки нового пакета.

- **Group**: - указание высокоуровневым программам установки (таким как glint) где расположить программу в их иерархических структурах.
- **%description**: - полное описание пакета.
- **%prep** - Это второй раздел в spec-файле. Он используется для того, чтобы сделать исходные тексты готовыми к построению. Как правило, здесь происходит распаковка исходников и наложение соответствующих патчей.
- **%build** - В данном разделе располагаются команды, которые необходимо выполнить для построения программного обеспечения, после того, как оно было распаковано и были наложены соответствующие патчи.
- **%install** - В данном разделе располагаются команды необходимые для установки пакета.

### Скрипты, выполняемые до и после установки/удаления пакета

- **%pre** - макрос для выполнения предустановочного скрипта.
- **%post** - макрос для выполнения послеустановочного скрипта.
- **%preun** - макрос для выполнения скрипта перед удалением пакета.
- **%postun** - макрос для скрипта выполняемого после удаления пакета.
- **%files** - Здесь располагается список файлов, которые будут включены в бинарный rpm пакет.

Рассмотрим часть spec файла postfix, надо заметить, что у postfix так сказать нестандартный spec файл.

```
#
# /usr/src/redhat/SPECS/postfix.spec
#

# Собрать с поддержкой LDAP
%define LDAP 2

# Поддержка MySQL, для того, чтобы ее включить, необходимо поменять 0 на 1
%define MYSQL 0

# Включить поддержку PCRE
%define PCRE 1

# Включить поддержку SASL2
%define SASL 2

# Включить поддержку TLS
%define TLS 1

# Включить поддержку протокола ipv6
%define IPV6 1

# Задает gid группы postdrop
%define POSTDROP_GID 90

# Собрать пакет pflogsum (анализатор лог файлов postfix)
%define PFLOGSUMM 1
```

```
...
...
...

# Здесь указываются sources (источники) исходных кодов, а также различных
# конфигурационных файлов и скриптов. Все эти файлы должны располагаться
# в папке /usr/src/redhat/SOURCES
Source0: ftp://ftp.porcupine.org/mirrors/postfix-release/official/
%{name}-%{version}.tar.gz
Source1: postfix-etc-init.d-postfix
Source3: README-Postfix-SASL-RedHat.txt
Source53:
http://jimsun.linxnet.com/downloads/pflogsumm-%{pflogsumm_ver}.tar.gz
Source100: postfix-sasl.conf
Source101: postfix-pam.conf

# Различные патчи, все патчи также должны располагаться в /usr/src/redhat/SOURCES
# Если вам необходимо применить патч, который отсутствует в данном списке,
# то необходимо его указать в этом списке, например
# Patch10: postfix-2.1.1-mypatch.patch. Учтите, что в некоторых патчах имеет
# значения порядок их наложения, т.е. например patch5 необходимо применить
# перед патчем 7 или 3. Яркий пример патчи для ядра, если у вас есть
# исходники ядра 2.6.12 и есть патчи 2.6.13 и 2.6.14. То вы не сможете сразу
# наложить патч 2.6.14, сначала надо наложить 2.6.13, а затем 2.6.14.

# Patches
Patch1: postfix-2.1.1-config.patch
Patch3: postfix-alternatives.patch
Patch4: postfix-hostname-fqdn.patch
Patch6: postfix-2.1.1-obsolete.patch
Patch7: postfix-2.1.5-aliases.patch
Patch8: postfix-large-fs.patch
Patch9: postfix-2.2.5-cyrus.patch
Patch10: postfix-2.1.1-mypatch.patch

# Базовый набор пакетов, необходимый для сборки postfix. Если у вас не
# будет установлен какой-либо из этих пакетов, то при попытке сборки вы
# получите соответствующее сообщение.
BuildRequires: gawk, perl, sed, ed, db4-devel, pkgconfig, zlib-devel
Requires: setup >= 2.5.36-1
BuildRequires: setup >= 2.5.36-1

...
...
...

# Если вы хоть немного разбираетесь в программировании, то думаю вам
# будет понятно для чего это. Т.е. здесь производится различная проверка.

# Например, если мы определили переменную LDAP (%define LDAP 2), т.е.
# мы хотим собрать postfix с поддержкой LDAP, то для сборки rpm нам
```

```
# понадобятся два пакета openldap и openldap-devel. Данная запись
# openldap >= 2.0.27 и openldap-devel >= 2.0.27 означает, что для
# сборки postfix с LDAP необходимо, чтобы в системе были установлены
# два пакета openldap и openldap-devel версии не ниже 2.0.27. Причем
# обратите внимание, что для работы postfix (Requires: openldap >= 2.0.27)
# необходим только один пакет openldap, что является логичным, так как
# devel пакеты необходимы только для сборки пакетов.
%if %{LDAP}
BuildRequires: openldap >= 2.0.27, openldap-devel >= 2.0.27
Requires: openldap >= 2.0.27
%endif

%if %{SASL}
BuildRequires: cyrus-sasl >= 2.1.10, cyrus-sasl-devel >= 2.1.10
Requires: cyrus-sasl >= 2.1.10
%endif

%if %{PCRE}
Requires: pcre
BuildRequires: pcre, pcre-devel
%endif

%if %{MYSQL}
Requires: mysql
BuildRequires: mysql, mysql-devel
%endif

%if %{TLS}
Requires: openssl
BuildRequires: openssl-devel >= 0.9.6
%endif
```

Например, мы захотели собрать postfix с поддержкой MySQL, для этого в самом начале меняем **%define MYSQL 0** на **%define MYSQL 1**. и снова выполняем команду

```
# rpmbuild -ba --target=i686 postfix.spec
Платформы для сборки: i686
Сборка для платформы i686
ошибка: Неудовлетворенные зависимости сборки:
        mysql-devel нужен для postfix-2.2.8-1.2.i686
```

Он нам пишет, что для сборки необходимо установить пакет mysql-devel. Обратите внимание, что версия не указывается, это значит, что можно установить любую версию, которую поддерживает postfix или нужный вам пакет.

Если бы вы собирали из исходных кодов, то вам пришлось бы самому искать, какие пакеты необходимы для сборки данного пакета. В этом и заключается одно из преимуществ сборки из src.rpm по сравнению с tar.gz или tar.bz2.

Устанавливаем соответствующий пакет

```
# rpm -ivh MySQL-devel-4.1.9-0.i386.rpm
Подготовка... #####
[100%]
  1:MySQL-devel #####
[100%]
```

И заново запускаем сборку postfix. На этот раз мы видим, что все необходимые пакеты для сборки установлены и теперь необходимо, лишь дождаться окончания сборки.

```
# rpmbuild -ba --target=i686 postfix.spec
Платформы для сборки: i686
Сборка для платформы i686
Выполняется(%prep): /bin/sh -e /var/tmp/rpm-tmp.86320
+ umask 022
+ cd /usr/src/redhat/BUILD
+ umask 022
+ cd /usr/src/redhat/BUILD
+ rm -rf postfix-2.2.8

...
...
...

Записан: /usr/src/redhat/SRPMS/postfix-2.2.8-1.2.src.rpm
Записан: /usr/src/redhat/RPMS/i686/postfix-2.2.8-1.2.i686.rpm
Записан: /usr/src/redhat/RPMS/i686/postfix-pflogsumm-2.2.8-1.2.i686.rpm
Выполняется(%clean): /bin/sh -e /var/tmp/rpm-tmp.52381
+ umask 022
+ cd /usr/src/redhat/BUILD
+ cd postfix-2.2.8
+ /bin/rm -rf /var/tmp/postfix-buildroot
+ exit 0
```

Все пакет у нас собран, теперь необходимо установить его и радоваться жизни.

```
# rpm -ivh /usr/src/redhat/RPMS/i686/postfix-2.2.8-1.2.i686.rpm
Подготовка... #####
[100%]
  1:postfix #####
[100%]
```

Для лучшего понимания рассмотрим сборку squid, который имеет более стандартную структуру спес файла. Как всегда для начала устанавливаем src.rpm, при этом не забываем проверить размер и контрольную сумму.

```
# rpm -ivh squid-2.5.STABLE11-2.src.rpm
  1:squid #####
[100%]
```

```
#
```

```
# /usr/src/redhat/SPECS/squid.spec
#
# Для добавления или удаления той или иной возможности здесь необходимо
# указать ключи сборки, которые используются в ./configure. Собственно вся
# настройка сводится к редактированию секции %configure
%configure \
  --exec_prefix=/usr \
  --bindir=%{_sbindir} \
  --libexecdir=%{_libdir}/squid \
  --localstatedir=/var \
  --sysconfdir=/etc/squid \
  --enable-poll \
  --enable-snmp \
  --enable-removal-policies="heap,lru" \
  --enable-storeio="aufs,coss,diskd,null,ufs" \
  --enable-delay-pools \
  --enable-linux-netfilter \
  --with-pthreads \
  --enable-ntlm-auth-helpers="winbind" \
  --enable-basic-auth-helpers="NCSA,winbind" \
  --enable-external-acl-helpers="ip_user,wbinfinfo_group,winbind_group" \
  --enable-auth="basic,ntlm" \
  --with-winbind-auth-challenge \
  --enable-useragent-log \
  --enable-referer-log \
  --disable-dependency-tracking \
  --enable-cachemgr-hostname=localhost \
  --disable-ident-lookups \
  --enable-truncate \
  --enable-underscores \
  --datadir=%{_datadir} \
  --enable-icmp --enable-htcp --enable-arp-acl \
  --enable-default-err-language="Russian-1251"
```

Узнать все возможные ключи можно следующим образом.

```
# cd /usr/src/redhat/SPECS
# rpmbuild --bp squid.spec
# cd ../BUILD/squid-2.5.STABLE11/
# ./configure --help
Usage: configure [options] [host]
Options: [defaults in brackets after descriptions]
Configuration:
  --cache-file=FILE      cache test results in FILE
  --help                 print this message
  --no-create            do not create output files
  --quiet, --silent     do not print `checking...' messages
  --site-file=FILE      use FILE as the site file
  --version              print the version of autoconf that created
configure
```



## Directory and file names:

```
--prefix=PREFIX      install architecture-independent files in PREFIX
                       [/usr/local/squid]
--exec-prefix=EPREFIX install architecture-dependent files in EPREFIX
                       [same as prefix]
--bindir=DIR          user executables in DIR [EPREFIX/bin]
--sbindir=DIR         system admin executables in DIR [EPREFIX/sbin]
--libexecdir=DIR     program executables in DIR [EPREFIX/libexec]
--datadir=DIR         read-only architecture-independent data in DIR
                       [PREFIX/share]
--sysconfdir=DIR     read-only single-machine data in DIR [PREFIX/etc]
--sharedstatedir=DIR modifiable architecture-independent data in DIR
                       [PREFIX/com]
--localstatedir=DIR  modifiable single-machine data in DIR [PREFIX/var]
--libdir=DIR          object code libraries in DIR [EPREFIX/lib]
--includedir=DIR     C header files in DIR [PREFIX/include]
--oldincludedir=DIR  C header files for non-gcc in DIR [/usr/include]
--infodir=DIR        info documentation in DIR [PREFIX/info]
--mandir=DIR         man documentation in DIR [PREFIX/man]
--srcdir=DIR          find the sources in DIR [configure dir or ..]
--program-prefix=PREFIX prepend PREFIX to installed program names
--program-suffix=SUFFIX append SUFFIX to installed program names
--program-transform-name=PROGRAM
                       run sed PROGRAM on installed program names
```

## Host type:

```
--build=BUILD        configure for building on BUILD [BUILD=HOST]
--host=HOST           configure for HOST [guessed]
--target=TARGET       configure for TARGET [TARGET=HOST]
```

## Features and packages:

```
--disable-FEATURE    do not include FEATURE (same as --enable-
FEATURE=no)
--enable-FEATURE[=ARG] include FEATURE [ARG=yes]
--with-PACKAGE[=ARG]  use PACKAGE [ARG=yes]
--without-PACKAGE     do not use PACKAGE (same as --with-PACKAGE=no)
--x-includes=DIR      X include files are in DIR
--x-libraries=DIR     X library files are in DIR
--enable and --with options recognized:
--disable-dependency-tracking Speeds up one-time builds
--enable-dependency-tracking Do not reject slow dependency extractors
--enable-maintainer-mode enable make rules and dependencies not useful
                          (and sometimes confusing) to the casual installer
--enable-dlmalloc[=LIB] Compile & use the malloc package by Doug Lea
--enable-gnuregex     Compile GNUregex. Unless you have reason to use
this
                          option, you should not enable it. This library
file
                          is usually only required on Windows and very old
                          Unix boxes which do not have their own regex
library
                          built in.
--enable-xmalloc-statistics
```

```

        Show malloc statistics in status page
--enable-carp          Enable CARP support
--enable-async-io[=N_THREADS]
                        Shorthand for
                        --with-aufs-threads=N_THREADS
                        --with-pthreads
                        --enable-storeio=ufs,aufs
--with-aufs-threads=N_THREADS
                        Tune the number of worker threads for the aufs
object
                        store.
--with-pthreads        Use POSIX Threads
--with-aio              Use POSIX AIO
--with-dl               Use dynamic linking
--enable-storeio="list of modules"
                        Build support for the list of store I/O modules.
                        The default is only to build the ufs module.
                        See src/fs for a list of available modules, or
                        Programmers Guide section
                        for details on how to build your custom store
module
--enable-heap-replacement
                        Backwards compatibility option. Please use the
                        new --enable-removal-policies directive instead.
--enable-removal-policies="list of policies"
                        Build support for the list of removal policies.
                        The default is only to build the lru module.
                        See src/repl for a list of available modules, or
                        Programmers Guide section 9.9 for details on how
                        to build your custom policy
--enable-icmp          Enable ICMP pinging
--enable-delay-pools   Enable delay pools to limit bandwidth usage
--enable-useragent-log Enable logging of User-Agent header
--enable-referer-log   Enable logging of Referer header
--disable-wccp         Disable Web Cache Coordination Protocol
--enable-kill-parent-hack
                        Kill parent on shutdown
--enable-snmp          Enable SNMP monitoring
--enable-cachemgr-hostname[=hostname]
                        Make cachemgr.cgi default to this host
--enable-arp-acl       Enable use of ARP ACL lists (ether address)
--enable-htcp          Enable HTCP protocol
--enable-ssl           Enable ssl gatewaying support using OpenSSL
--with-openssl[=prefix]
                        Compile with the OpenSSL libraries. The path to
                        the OpenSSL development libraries and headers
                        installation can be specified if outside of the
                        system standard directories
--enable-forw-via-db   Enable Forw/Via database
--enable-cache-digests
                        Use Cache Digests
                        see http://www.squid-cache.org/FAQ/FAQ-16.html

```

```

--enable-default-err-language=lang
    Select default language for Error pages (see
    errors directory)
--enable-err-languages="lang1 lang2.."
    Select languages to be installed. (All will be
    installed by default)
--with-coss-membuf-size COSS membuf size (default 1048576 bytes)
--enable-poll
    Enable poll() instead of select(). Normally poll
    is preferred over select, but configure knows poll
    is broken on some platforms. If you think you are
    smarter than the configure script, you may enable
    poll with this option.
--disable-poll
    Disable the use of poll().
--disable-http-violations
    This allows you to remove code which is known to
    violate the HTTP protocol specification.
--enable-ipf-transparent
    Enable Transparent Proxy support for systems
    using IP-Filter network address redirection.
--enable-pf-transparent
    Enable Transparent Proxy support for systems
    using PF network address redirection.
--enable-linux-netfilter
    Enable Transparent Proxy support for Linux 2.4.
--with-large-files
    Enable support for large files (logs etc).
--enable-large-cache-files
    Enable support for large cache files (>2GB).
    WARNING: on-disk cache format is changed by this
option
  --with-build-environment=model
    The build environment to use. Normally one of
    POSIX_V6_ILP32_OFF32  32 bits
    POSIX_V6_ILP32_OFFBIG 32 bits with large file
support
    POSIX_V6_LP64_OFF64  64 bits
    POSIX_V6_LPBIG_OFFBIG large pointers and files
    XBS5_ILP32_OFF32     32 bits (legacy)
    XBS5_ILP32_OFFBIG    32 bits with large file
suppor
    XBS5_LP64_OFF64     64 bits (legacy)
    XBS5_LPBIG_OFFBIG   large pointers and files
    default              The default for your OS
--enable-leakfinder
    Enable Leak Finding code. Enabling this alone
    does nothing; you also have to modify the source
    code to use the leak finding functions. Probably
    Useful for hackers only.
--disable-ident-lookups
    This allows you to remove code that performs
    Ident (RFC 931) lookups.
--disable-internal-dns
    This prevents Squid from directly sending and

```

```

receiving DNS messages, and instead enables the
old external 'dnsserver' processes.
--enable-truncate This uses truncate() instead of unlink() when
removing cache files. Truncate gives a little
performance improvement, but may cause problems
when used with async I/O. Truncate uses more
filesystem inodes than unlink..
--disable-hostname-checks Squid by default rejects any host names with
odd characters in their name to conform with
internet standards. If you disagree with this
you may use this switch to turn off any such
checks, provided that the resolver used by
Squid does not reject such host names.. This
may be required to participate in testbeds for
international domain names.
--enable-underscores Squid by default rejects any host names with _
in their name to conform with internet standards.
If you disagree with this you may allow _ in
hostnames by using this switch, provided that
the resolver library on the host where Squid runs
does not reject _ in hostnames...
--enable-auth="list of auth scheme modules"
Build support for the list of authentication
schemes.
The default is to build support for the Basic
scheme.
See src/auth for a list of available modules, or
Programmers Guide section authentication schemes
for details on how to build your custom auth
scheme
module
--enable-auth-modules="list of helpers"
Backwards compatibility alias for
--enable-basic-auth-helpers
--enable-basic-auth-helpers="list of helpers"
This option selects which basic scheme proxy_auth
helpers to build and install as part of the normal
build process. For a list of available
helpers see the helpers/basic_auth directory.
--enable-ntlm-auth-helpers="list of helpers"
This option selects which proxy_auth ntlm helpers
to build and install as part of the normal build
process. For a list of available helpers see
the helpers/ntlm_auth directory.
--enable-digest-auth-helpers="list of helpers"
This option selects which digest scheme
authentication
helpers to build and install as part of the normal
build
process. For a list of available helpers see the

```

```

helpers/digest_auth directory.
--enable-ntlm-fail-open Enable NTLM fail open, where a helper that fails
                        one of the Authentication steps can allow squid to
                        still authenticate the user.
--enable-external-acl-helpers="list of helpers"
                        This option selects which external_acl helpers to
                        build and install as part of the normal build
                        process. For a list of available helpers see the
                        helpers/external_acl directory.
--with-samba-sources=/path/to/samba-source-tree
                        Path where the correct Samba source files can be
                        found while building winbind helpers. (defaults to
                        use internal copies of the headers from
Samba-2.2.7)

--disable-unlinkd      Do not use unlinkd
--enable-stacktraces  Enable automatic call backtrace on fatal errors
--enable-x-accelerator-vary
                        Enable support for the X-Accelerator-Vary
                        HTTP header. Can be used to indicate
                        variance within an accelerator setup.
                        Typically used together with other code
                        that adds custom HTTP headers to the requests.
--with-maxfd=N        Override maximum number of filedescriptors. Useful
                        if you build as another user who is not privileged
                        to use the number of filedescriptors you want the
                        resulting binary to support

```

После того, как вы нашли необходимый ключ, добавляем его в %configure. Например, мы хотим собрать squid с поддержкой ssl. Из помощи мы определили, что для этого, необходимо добавить два ключа **-enable-ssl** и **-with-openssl**. Вносим соответствующие изменения

```

#
# /usr/src/redhat/SPECS/squid.spec
#

%configure \
  --exec_prefix=/usr \
  --bindir=%{_sbindir} \
  --libexecdir=%{_libdir}/squid \
  --localstatedir=/var \
  --sysconfdir=/etc/squid \
  --enable-poll \
  ...
  ...
  ...
  --enable-ssl \
  --with-openssl \
  ...
  ...
  ...

```

```
--datadir=%{_datadir} \  
--enable-icmp --enable-htcp --enable-arp-acl \  
--enable-default-err-language="Russian-1251"
```

Сохраняем файл и начинаем сборку.

```
# rpmbuild -ba --target=athlon squid.spec  
Платформы для сборки: athlon  
Сборка для платформы athlon  
Выполняется(%prep): /bin/sh -e /var/tmp/rpm-tmp.59199  
+ umask 022  
+ cd /usr/src/redhat/BUILD  
+ cd /usr/src/redhat/BUILD  
+ rm -rf squid-2.5.STABLE11  
+ /usr/bin/bzip2 -dc /usr/src/redhat/SOURCES/squid-2.5.STABLE11.tar.bz2  
  
...  
...  
...  
  
SSL gatewaying using OpenSSL enabled  
Using OpenSSL MD5 implementation  
  
...  
...  
...  
  
Записан: /usr/src/redhat/SRPMS/squid-2.5.STABLE11-2.src.rpm  
Записан: /usr/src/redhat/RPMS/athlon/squid-2.5.STABLE11-2.athlon.rpm  
Выполняется(%clean): /bin/sh -e /var/tmp/rpm-tmp.7322  
+ umask 022  
+ cd /usr/src/redhat/BUILD  
+ cd squid-2.5.STABLE11  
+ rm -rf /var/tmp/squid-2.5.STABLE11-root  
+ exit 0  
Выполняется(--clean): /bin/sh -e /var/tmp/rpm-tmp.7322  
+ umask 022  
+ cd /usr/src/redhat/BUILD  
+ rm -rf squid-2.5.STABLE11  
+ exit 0
```

Все squid у нас собран успешно, теперь осталось только установить или обновить его.

From:  
<http://sys-adm.org.ua/> - [wiki.sys-adm.org.ua](http://wiki.sys-adm.org.ua)

Permanent link:  
<http://sys-adm.org.ua/system/rpm-build>

Last update: **2009/09/05 21:06**



