

YUM

Программа Yum облегчает управление пакетами. Она основана на RPM и предоставляет множество дополнительных функций.

В качестве источников данных (репозиториев) служат архивы Yum, расположенные в Интернете. Yum-архив — это собрание пакетов RPM, дополнительно снабженных метаданными (каталог repodata). В метаданных находится информация о содержимом и зависимостях всех пакетов. Благодаря разделению пакетов и метаданных обработка информации ускоряется (так как не нужно считывать все пакеты, чтобы извлечь данные). Yum способна автоматически переходить между несколькими зеркала, на которых расположен источник пакетов.

Yum автоматически отменяет все взаимозависимости пакетов, загружает все необходимые пакеты и устанавливает их. При этом учитываются все источники пакетов. Если вы, например, устанавливаете пакет из репозитория А, может быть, что Yum предварительно загрузит определенные связанные с ним пакеты из источников В и С.

Программа Yum может обновить все установленные пакеты одной-единственной командой. Для этого она проверяет, имеются ли в доступных источниках обновленные версии установленных пакетов. Если это так, то необходимые пакеты скачиваются и устанавливаются. Разумеется, при этом снимаются и все взаимозависимости пакетов.

Yum была разработана на языке программирования Python. Программа по умолчанию применяется в том числе в Fedora и в Red Hat. В этом разделе будет рассмотрено применение Yum в Fedora 11.

Конфликты с блокировкой

Нельзя одновременно запускать несколько образцов Yum. Если уже работает программа или команда Yum, то при повторном запуске выводится сообщение об ошибке `another copy is running` (работает другой экземпляр).

Суть проблемы заключается в работе системы автоматических обновлений Yum. Этот процесс несложно обнаружить: файл `/var/run/yum.pid` содержит ID-номер программы, вызывающей конфликт. С помощью команды `ps grep id` можно узнать имя программы. Если необходимо, можно временно остановить систему обновлений с помощью команды `/etc/init.d/yum-updatesd stop`, а затем вновь запустить ее командой `start`.

Конфигурация

Основная конфигурация Yum выполняется в файле `/etc/yum.conf`. В следующих строках показаны выдержки из конфигурации в системе Fedora 12:

```
# Файл /etc/yum.conf [main]

cachedir=/var/cache/yum/$basearch/$releasever
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit=3
```

Кратко опишу важнейшие настройки. Благодаря кеерскаше=0 скачанные пакеты не архивируются после установки. Как правило, эта установка полезна, так как дисковое пространство, необходимое для пакетов, со временем сильно вырастет, и, скорее всего, не будет причин устанавливать пакеты повторно. В любом случае Yum может обнаружить в ходе установки проблему и прервать установку. Если есть возможность устранить проблему и повторно произвести обновления, все пакеты потребуется скачать заново. Чтобы избежать этого, укажите кеерскаше=1. Для того чтобы специально удалить пакеты, находящиеся в /var/cache/yum, выполните команду yum clean packages.

При наличии exaktarch=1 Yum учитывает только такие обновления, архитектура которых соответствует архитектуре уже установленного пакета. Иначе говоря, заменить пакеты i386 новыми пакетами x86_64 не получится.

При использовании gpgcheck=1 Yum проверяет ключ аутентификации пакета. Переменная gpgcheck также можно настроить иначе, чем указано в yum.conf, — отдельно для каждого репозитория. Переменная plugins определяет, будет ли Yum учитывать плагины.

Некоторые пакеты Yum должна установить, но не должна обновлять. К ним относятся, в частности, пакеты ядра: при обновлении ядра дополнительно устанавливается новый пакет ядра, не затрагивая при этом старый пакет. Переменная installonlypkgs позволяет указывать названия таких пакетов. По умолчанию она имеет настройки kernel, kernel-smp, kernel-bigmem, kernel-enterprise, kernel-debug, kernel-unsupported. Наконец, переменная installonly_limit, содержащаяся в yum.conf, указывает, сколько версий подобных пакетов может быть установлено параллельно. При стандартной настройке одновременно остаются установлены только три новейшие версии пакета. Более старые пакеты ядра удаляются.

Создание репозитория. Каждый репозиторий определяется в собственном REPO-файле в каталоге /etc/yum.repos.d. В следующем фрагменте показаны репозитории для получения основных пакетов Fedora 12:

```
# Файл /etc/yum.repos.d/<a class="myClass" href="http://www.modx.cc/linux/zapusk-sistemyi-v-fedora/">fedora</a>.repo [fedora]
```

```
name=Fedora $releasever - $basearch
failovermethod=priority
mirrorlist=
https://mirrors.fedoraproject.org/metalink?repo=fedora-$releasever&arch=$basearch enabled=1
metadata_expire=7d gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch
```

Адрес репозитория можно по выбору указать в абсолютной форме с помощью baseurl =... или с использованием mirrorlist=... в виде файла-зеркала. В этом файле расположен список зеркальных серверов. Yum самостоятельно выбирает одно из зеркал. В конфигурационном файле Yum заменяет переменные \$releasever, \$arch и \$basearch номерами версии дистрибутива Linux с указанием ее архитектуры. Кратко рассмотрим происхождение этих трех переменных.

— \$arch определяется с помощью команды uname (если быть точным — с помощью одноименной функции Python, основанной на uname) и возвращает, к примеру, x86_64, если мы имеем дело с процессором 64-Bit-Intel/AMD.

— \$basearch — это базовая архитектура, лежащая в основе \$arch (например, i386).

— \$releasever получается из номера версии пакета edhat-release или fedora-release (также можно указать в yum.conf название другого пакета с помощью ключевого слова distroverpkg; номер версии этого пакета считается номером версии дистрибутива).

Переменная metadata_expires указывает, в течение какого срока остаются действительны скачанные метаданные. Yum сохраняет метаданные в кэше и не скачивает их повторно, если прежние метаданные еще не устарели. Так экономится время и объем закачек (некоторые файлы с метаданными просто огромны), но в этом случае Yum может проигнорировать изменения, недавно

внесенные в репозиторий (при необходимости можно удалить локальные метаданные командой `yum clean metadata`, тогда при следующем запуске `Yum` придется заново считать метаданные всех пакетов).

Оптимальная настройка для `metadata_expires` отличается в зависимости от репозитория: если его содержимое обновляется редко или вообще не обновляется, лучше задать длительный промежуток. Напротив, при работе с пакетами обновлений желательно задать короткий период или вообще отказаться от этой настройки.

Запрет пакетов. Если вы хотите, чтобы `Yum` не контактировала с определенными пакетами и при появлении новой версии они также не обновлялись, вставьте в `yum.conf` или в файл `*.repo` репозитория строку вида `exclude имя1 имя2 имя3`. В названиях пакетов можно использовать джокерные символы, то есть допускается и такая запись: `exclude хеmас*`.

Примеры

Управление пакетами целиком осуществляется с помощью команды `yum`. Синтаксис этой команды понятен из следующего примера. Приведенные далее команды показывают, как работает программа (вывод сокращен ради экономии места).

```
root# yum check-update
```

```
2.3-75.fc11 updates
```

```
2.26.2-1.fc11 updates
```

```
2.26.2-1.fc11 updates
```

```
2.26.2-1.fc11 updates
```

```
2.26.2-1.fc11 updates
```

```
1.3-1.fc11 updates
```

```
апасропл'586 brasero.i586
```

```
brasero-libs.i586 brasero-nautilus.i586
```

```
cheese.i586 cronie.i586
```

```
root# yum update
```

```
Transaction Summary Install 1 Package(s) Update 42 Package(s) Remove 0 Package(s) Total download size: 52 M Is this ok [y/N]: y
```

```
root# yum install emacs
```

```
Installing:
```

```
emacs i586 l:22.3-12.fc11 updates 2.0M
```

```
Installing for dependencies: emacs-common i 586 l:22.3-12.fc11 updates 19 M
```

```
xorg-x11-fonts-IS08859-1-100dpi noarch 7.2-8.fc11 <a class="myClass" href="http://www.modx.cc/linux/zapusk-sistemyi-v-fedora/">fedora</a> 1.1 M Transaction Summary Install 3 Package(s) Update 0 Package(s) Remove 0 Package(s) Total download size: 22 M Is this ok [y/N]: y
```

Если вы впервые выполняете команду `yum`, система скачивает метаинформацию по всем созданным репозиториям. В дальнейшем эти файлы регулярно обновляются.

Группы пакетов

Программа `Yum` может распределять пакеты по группам, чтобы проще было установить все пакеты, необходимые для решения определенной задачи. Список доступных групп пакетов выводится командой `yum grouplist`, а `yum groupinfo name` подсказывает, какие пакеты относятся к одной и той же группе. Команда `yum groupinfo` подразделяет все пакеты на три категории: `mandatory` (обязательные), `default` (по умолчанию) и `optional` (необязательные). Выполнив `yum groupinstall имя`, вы установите все пакеты групп `mandatory` и `default`. Команда `yum` не содержит параметров, которые позволили бы автоматически установить и необязательные пакеты. Если вам нужна такая возможность, внесите в файл `yum.conf` следующие изменения:

```
# Дополнение в /etc/yum.conf group_package_types = mandatory default optional
```

Если хотите обновить или удалить группу пакетов, используйте `yum groupupdate` или `yum groupremove` соответственно.

Пакеты с исходным кодом

Сама по себе команда `yum` не может устанавливать пакеты с исходным кодом. Эту задачу выполняет команда `yumdownloader` из пакета `yum-utils`. Следующая команда загружает пакет с исходным кодом редактора `gedit` в локальный каталог. При этом активизируются источники `source`, находящиеся в файлах `*.repo` и по умолчанию неактивные.

```
user$ yumdownloader --source gedit
```

Автоматические загрузки и обновления

Если установлен пакет `yum-updatesd`, то при запуске компьютера процесс `Init-V` активизирует одноименную программу `yum-updatesd`.

Команда `yum-updatesd` позволяет регулярно проверять, есть ли доступные обновления. В зависимости от того, какие настройки сделаны в `/etc/yum/yum-updatesd.conf`, пакеты обновлений также загружаются и даже устанавливаются автоматически. В следующем листинге показана стандартная конфигурация Fedora 11, при которой автоматические обновления не производятся. Если вы хотите производить автоматические обновления, в трех случаях измените `no` на `yes`:

```
# /etc/yum/yum-updatesd.conf [main]
# Проверять один раз в час, доступны ли новые обновления run_interval = 3600
# Связываться с сервером обновлений не реже, чем каждые 10 минут updaterefresh = 600
# Осуществлять локальное оповещение об уведомлениях через dbus emit_via = dbus
# Автоматическая загрузка обновлений do_download = no
# Загрузить для обновления зависимые пакеты do_download_deps = no
# Автоматически установить обновления do_update = no
```

К сожалению, если программа обновлений `Yum` будет работать постоянно, это иногда может приводить к проблемам, связанным с блокировкой (см. подраздел «Конфигурация» этого раздела). Временно остановите `yum-updatesd`, прежде чем приступить к управлению пакетами. Не забудьте потом снова запустить `yum-updatesd`!

```
root# /etc/init.d/yum-updatesd stop
root# ... вручную установить или удалить пакеты ...
root# /etc/init.d/yum-updatesd start
```

Install the EPEL repository

You install the EPEL repository by downloading the appropriate RPM package for your system and installing it. The following instructions use the 64-bit packages that work with Rackspace Cloud Servers instances.

CentOS Extras repository

The CentOS Extras repository includes a package to install EPEL, and is enabled by default. To install the EPEL package, run the following command:

```
sudo yum install epel-release
```

If that command doesn't work, perhaps because the CentOS Extras repository is disabled, use the following manual installation instructions based on your distribution version:

- CentOS and Red Hat Enterprise Linux 6.x

```
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
sudo rpm -Uvh epel-release-6*.rpm
```

- CentOS and Red Hat Enterprise Linux 7.x

```
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
sudo rpm -Uvh epel-release-latest-7*.rpm
```

If you get a `File Not Found` error message when trying to download the package, the version number might have changed. You can access the latest version of the RPM installer from the [Fedora EPEL wiki page](#). The wiki page also includes additional instructions for Red Hat Network subscribers who are installing the EPEL repository.

Install the IUS repository

The IUS repository provides newer versions of some software in the official CentOS and Red Hat repositories. The IUS repository depends on the EPEL repository.

The package names in the IUS repository are *different from* the package names used in the official repositories. The difference helps to avoid unintentional conflicts or software version updates.

Note: Because IUS uses package names that are different from the package names in the official repositories, we recommend IUS over Remi for Rackspace customers with managed support levels that include server software. Managed servers automatically update nightly by default, which can cause unplanned upgrades if package names are the same in more than one enabled repository.

You install the IUS repository by downloading the appropriate RPM package for your system and installing it. The following instructions use the 64-bit packages that work with Cloud Servers instances.

- CentOS 6.x

```
wget https://centos6.iuscommunity.org/ius-release.rpm  
sudo rpm -Uvh ius-release*.rpm
```

- CentOS 7.x

```
wget https://centos7.iuscommunity.org/ius-release.rpm  
sudo rpm -Uvh ius-release*.rpm
```

- Red Hat 6.x

```
wget https://rhel6.iuscommunity.org/ius-release.rpm  
sudo rpm -Uvh ius-release*.rpm
```

- Red Hat 7.x

```
wget https://rhel7.iuscommunity.org/ius-release.rpm  
sudo rpm -Uvh ius-release*.rpm
```

If you get a `File Not Found` error message when trying to download the package, the version number might have changed. You can access the latest version of the repository package from the [IUS Getting Started Page](#).

Upgrade installed packages to IUS versions

If you already have a software package installed that you want to upgrade to a newer version in the IUS repository, install the IUS yum plug-in for package replacement to simplify the upgrade process.

```
sudo yum install yum-plugin-replace
```

The plug-in provides a `yum replace` command that replaces a specified package and installs any required dependencies at the same time. For example, to replace the installed PHP package with the PHP 5.6 package from the IUS repository, run the following command:

```
sudo yum replace php --replace-with php56u
```

For more information, see the [IUS Usage Guide](#).

Install the Remi repository

The Remi repository provides newer versions of the software in the core CentOS and Red Hat Enterprise Linux repositories. The Remi repository depends on the EPEL repository.

Package names in the Remi repository are the same as the package names used in the official repositories. This similarity can result in inadvertent package upgrades when you run an update with yum, so use the Remi repository with care.

Note: Because Remi uses package names that are the same as the package names in the official repositories, we do not recommend Remi for Rackspace customers with a managed level of support. Managed servers automatically update nightly by default, which can cause unplanned upgrades if the Remi repository is enabled. If you require the Remi repository or another repository with

package name conflicts, contact Rackspace Support before applying any upgrades to ensure continued support for your server.

You install the Remi repository by downloading the appropriate RPM package for your system and installing it. The following instructions use the 64-bit packages that work with Cloud Servers instances.

- CentOS and Red Hat Enterprise Linux 6.x

```
wget http://rpms.famillecollet.com/enterprise/remi-release-6.rpm
sudo rpm -Uvh remi-release-6*.rpm
```

- CentOS and Red Hat Enterprise Linux 7.x

```
wget http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
sudo rpm -Uvh remi-release-7*.rpm
```

If you get a `File Not Found` error message when trying to download the package, the version number might have changed. You can access the latest version of the RPM installer from the [Remi Repository Configuration](#) page. The configuration page also includes additional instructions for Red Hat Network subscribers who are installing the Remi repository.

Enable the Remi repository

The Remi repository is disabled by default.

To use the Remi repository only when you know you need it, use the `--enablerepo=remi` option when installing a package. For example:

```
sudo yum --enablerepo=remi install php-tcpdf
```

If you want to permanently enable the Remi repository, you need to edit the yum configuration file for Remi.

Open the repository configuration file by using a text editor of your choice. This example uses nano.

```
sudo nano /etc/yum.repos.d/remi.repo
```

Edit the `[remi]` portion of the file to set the `enabled` option to 1. This action enables the Remi repository by default.

```
name=Les RPM de remi pour Enterprise Linux $releasever - $basearch
#baseurl=http://rpms.famillecollet.com/enterprise/$releasever/remi/$basearch/
mirrorlist=http://rpms.famillecollet.com/enterprise/$releasever/remi/mirror
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi
failovermethod=priority
```

Check for available repositories

You can see if the repositories that you need are installed and enabled by running the following command:

```
yum repolist
```

Some repositories, like Remi, are disabled by default. To list disabled repositories, run the following command:

```
yum repolist disabled
```