使用 **yum** 工具更新系统

内容提要

1. 理解为什么使用 yum
2. 了解 CentOS 的镜像站点目录结构
3. 理解 CentOS 的仓库及镜像站点的关系
4. 学会配置各种更新源
5. 学会使用 yum 命令工具
6. 学会为 yum 配置代理
7. 学会使用 yum-fastestmirror 加快更新
8. 理解系统自动更新的配置方法

yum 简介

为什么使用 yum

Linux系统维护中令管理员很头疼的就是软件包之间的依赖性了，往往是你要安装A软件，但是编译的时候告诉你X软件安装之前需要B软件，而当你安装Y软件的时候，又告诉你需要Z库了，好不容易安装好Z库，发现版本还有问题等。由于历史原因，RPM软件包管理系统对软件之间的依存关系没有内部定义，造成安装RPM软件时经常出现令人无法理解的软件依赖问题。其实开源社区早就对这个问题尝试进行解决了，不同的发行版推出了各自的工具，比如Yellow Dog的YUM（Yellow dog Updater, Modified），Debian的APT(Advanced Packaging Tool)等。开发这些工具的目的都是为了要解决安装RPM时的依赖性问题，而不是额外再建立一套安装模式。这些软件也被开源软件爱好者们逐渐移植到别的发行版上。目前，APT和YUM都可以运行在Red Hat系统上。目前 yum 是Red Hat/Fedora系统上默认安装的更新系统。

什么是 yum

yum（http://linux.duke.edu/projects/yum/ [http://www.proxyserve.net/index.php?q=aHR0cDovL2xpbnV4LmR1a2UuZWR1L3Byb2plY3RzL3I1bS8%3D]），是 Yellow dog Updater, Modified 的简称，起初是由yellow dog 发行版的开发者 Terra Soft 研发，用 python 写成，那时叫做 yup (yellow dog updater)，后经杜克大学的 Linux@Duke 开发团队进行改进，遂有此名。yum 的宗旨是自动化地升级，安装/移除rpm包，收集rpm包的相关信息，检查依赖性并自动提示用户解决。yum 的关键之处是要有可靠的 repository，顾名思义，这是软件的仓库，它可以是 http 或 ftp 站点，也可以是本地软件池，但必须包含 rpm 的 header，header 包括了rpm 包的各种信息，包括描述，功能，提供的文件，依赖性等.正是收集了这些 header并加以分析，才能自动化地完成余下的任务。

yum 具有如下特点：

- 自动解决包的倚赖性问题能更方便的添加/删除/更新RPM包
- 便于管理大量系统的更新问题
- 可以同时配置多个资源库(Repository)
- 简洁的配置文件(/etc/yum.conf)
- 保持与RPM数据库的一致性
- 有一个比较详细的log，可以查看何时升级安装了什么软件包等
- 使用方便

yum是CentOS/Fedora系统自带的，因此它能使用CentOS官方的软件源，完成各种官方发布的各种升级。对于第三方软件源的支持，yum也不差，大多数支持 apt 的 repository，也能支持 yum，比如说 freshrpms、fedora.us、livna、dag 等。

CentOS 的镜像和仓库

CentOS 的镜像站点

完整的 CentOS 软件库非常大。CentOS 的软件包位于 CentOS 的镜像站点 [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5jZW50b3Mub3JnL21vZHVsZXMvdGlueWNvbnRlbnQvaW5kZXgucGhwP2lkPTEz] 的目录树中，可通过 FTP 或 HTTP 访问它们。

- 在 http://mirror-status.centos.org/ [http://www.proxyserve.net/index.php?q=aHR0cDovL21pcnJvci1zdGF0dXMuY2VudG9zLm9yZy8%3D] 可以看到各镜像的状态。
- 在 http://www.centos.org/modules/tinycontent/index.php?id=22 [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5jZW50b3Mub3JnL21vZHVsZXMvdGlueWNvbnRlbnQvaW5kZXgucGhwP2lkPTIy] 可以看到如何进行镜像的知识。

镜像站点的第一级目录是发行版本号，如 3、4、5 等。 下列目录存在于任何 CentOS 镜像站点的版本号为5的目录下：

- **addons/**: 本目录包含 CentOS 的 addons 仓库
- **centosplus/**: 本目录包含 CentOS 的 centosplus 仓库
- **extras/**: 本目录包含 CentOS 的 extras 仓库

- **fasttrack/**: 本目录包含 CentOS 的 fasttrack 仓库
- **isos-dvd/**: 本目录包含发行版的 DVD iso 下载文件
- **isos/**: 本目录包含发行版的 CD isos 下载文件
- **os/**: 本目录包含发行版（distributions）的仓库
- **updates/**: 本目录包含 CentOS 的 updates 仓库

CentOS 的仓库

仓库（repository）是一个预备好的目录，或是一个网站，包含了软件包和索引文件。 yum 可以在仓库中自动地定位并获取正确的 RPM 软件包。这样，您就不必手动搜索和安装新应用程序和升级补丁了。只用一个命令，您就可以更新系统中所有软件，也可以根据指定搜索目标来查找安装新软件。

镜像服务器为每个版本的 CentOS 分别提供了一些仓库。CentOS 5 中的软件管理工具 yum 已经预先配置为使用下列四个仓库：

- **base**: 构成 CentOS 发行版（distributions）的软件包，和光盘上内容相同
- **updates**: base 仓库中软件包的更新版本
- **addons**: 已编译的但不在发行版（distributions）中的软件包
- **extras**: 一大批附加的软件包

还可以使用下面的仓库，但在 CentOS 5 默认情况下未被启用

- **centosplus/**: 用于增强一些现有软件包的功能

CentOS 仓库的目录结构

下面以 CentOS 5 的 **updates** 仓库为例说明其在镜像站点上的目录结构（其他仓库结构类似）：

```
updates/
|-- SRPMS                          # 源代码目录
|   |-- *.src.rpm                  # 源代码包文件
|   `-- repodata/                  # 源代码的索引文件
|-- i386                           # Intel 32位平台目录
|   |-- RPMS                       # Intel 32位平台的RPMS目录
|   |   |-- *.i386.rpm             # 在Intel 32位平台上编译的包文件
|   |   `-- *.centos.noarch.rpm    # 与平台无关的已编译的包文件
|   `-- repodata/                  # Intel 32位平台的索引文件
`-- x86_64                         # 64位平台目录
    |-- RPMS                       # 64位平台的RPMS目录
    |   |-- *.x86_64.rpm           # 在64位平台上编译的包文件
    |   `-- *.centos.noarch.rpm    # 与平台无关的已编译的包文件
    `-- repodata/                  # 64位平台的索引文件
```

配置 yum 客户的更新源

yum 使用 **reposdir** (/etc/yum.repos.d) 目录下的一系列 .repo 文件列出可获得软件包仓库的镜像站点地址。默认情况下有如下两个文件：

- CentOS-Base.repo ： 用于设置远程仓库
- CentOS-Media.repo ： 用于设置本地仓库

在 .repo 文件中，配置语法是分段的，每一段配置一个软件仓库，配置语法如下：

```
[repositoryid]
name=Some name for this repository
baseurl=url://server1/path/to/repository/
        url://server2/path/to/repository/
        url://server3/path/to/repository/
mirrorlist=url://path/to/mirrorlist/repository/
enabled=0/1
gpgcheck=0/1
gpgkey=A URL pointing to the ASCII-armoured GPG key file for the repository
```

其中：

- repositoryid ： 用于指定一个仓库
- name ： 用于指定易读的仓库名称
- baseurl ： 用于指定本仓库的 URL，可以是如下的几种类型：
  - http — 用于指定远程 HTTP 协议的源
  - ftp — 用于指定远程 FTP 协议的源
  - file — 用于本地镜像或 NFS 挂装文件系统
- mirrorlist ： 用于指定仓库的镜像站点
- enabled ： 用于指定是否使用本仓库，默认值为1，即可用
- gpgcheck ： 用于指定是否检查软件包的 GPG 签名
- gpgkey ： 用于指定 GPG 签名文件的 URL

在 name baseurl 中经常使用如下的变量：

- $releasever — 当前系统的版本号
- $basearch — 当前系统的平台架构

- 文件中以"#"开头的行是注释行
- 若指定 mirrorlist，系统将从 CentOS 的镜像站点中选择离您最近的仓库
- 并非所有的国内镜像都在 CentOS 的镜像站点列表中，所以我们可以直接使用 baseurl 直接指定离您最近的仓库
- baseurl 可以指定多个 URL，系统会依次检查您列出的仓库，以便在某个服务器宕机时可以使用另外的服务器
- 为了加快更新，在确保更新服务器及线路良好的情况下，在 baseurl 中只指定一个 URL 既可

设置网络更新源

下面是一个 CentOS-Base.repo 文件的实例，在此文件中没有设置 mirrorlist ，使用 baseurl 只指定了一个 URL。

```
[base]
name=CentOS-$releasever - Base
baseurl=http://centos.candishosting.com.cn/$releasever/os/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5

[updates]
name=CentOS-$releasever - Updates
baseurl=http://centos.candishosting.com.cn/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5

[addons]
name=CentOS-$releasever - Addons
baseurl=http://centos.candishosting.com.cn/$releasever/addons/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5

[extras]
name=CentOS-$releasever - Extras
baseurl=http://centos.candishosting.com.cn/$releasever/extras/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5

[centosplus]
name=CentOS-$releasever - Plus
baseurl=http://centos.candishosting.com.cn/$releasever/centosplus/$basearch/
gpgcheck=1
enabled=0
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5
```

设置本地更新源

为了使用安装光盘作为更新源，可以修改 CentOS-Media.repo 文件，下面是一个配置实例。

```
[c5-media]
name=CentOS-$releasever - Media
baseurl=file:///media/CentOS/
        file:///media/cdrom/
        file:///media/cdrecorder/
gpgcheck=1
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-beta
```

- 为了使用 c5-media 仓库，需将 CentOS-Base.repo 文件中的 base 仓库使用 enabled=0 设置成不可用。
- 若本地磁盘空间有足够空间，您也可以将安装光盘复制到本地磁盘的一个目录中

设置非官方更新源

下面以安装 rpmforge 软件包为例添加非官方更新源。

1. 到 http://dag.wieers.com/rpm/packages/rpmforge-release/ [http://www.proxyserve.net/index.php?q=aHR0cDovL2RhZy53aWVlcnMuY29tL3JwbS9wYWNrYWdlcy9ycG1mb3JnZS1yZWxlYXNlLw%3D%3D] 查找适用于 RHEL5/CentOS5 的软件包
2. 下载适用于 RHEL5/CentOS5 的软件包

```
#  wget http://dag.wieers.com/rpm/packages/rpmforge-release/rpmforge-release-0.3.6-1.el5.rf.i386.rpm
#  rpm -ivh rpmforge-release-0.3.6-1.el5.rf.i386.rpm
#  rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-rpmforge-*
```

安装了 rpmforge-release 软件包之后可以看到 /etc/yum.repos.d/rpmforge.repo 文件。以后使用 yum 就可以使用这个更新源了。

您也可以使用如下的手工方法添加 DAG 的仓库。

建立 dag.repo，定义非官方库：

```
#  vi /etc/yum.repos.d/dag.repo
```

```
[dag]
name=Dag RPM Repository for Red Hat Enterprise Linux
baseurl=http://apt.sw.be/redhat/el$releasever/en/$basearch/dag
gpgcheck=1
enabled=1
gpgkey=http://dag.wieers.com/packages/RPM-GPG-KEY.dag.txt
```

使用 yum 命令工具

yum 命令工具简介

yum [http://www.proxyserve.net/index.php?q=aHR0cDovL21hbi5jeC95dW0%3D] 完整语法参见其命令手册，下面只列出较常见的用法。

命令	功能
yum check-update	检查可更新的所有软件包
yum update	下载更新系统已安装的所有软件包
yum upgrade	大规模的版本升级,与yum update不同的是,连旧的淘汰的包也升级
yum install <packages>	安装新软件包
yum update <packages>	更新指定的软件包
yum remove <packages>	卸载指定的软件包
yum groupinstall <groupnames>	安装指定软件组中的软件包
yum groupupdate <groupnames>	更新指定软件组中的软件包
yum groupremove <groupnames>	卸载指定软件组中的软件包
yum grouplist	查看系统中已经安装的和可用的软件组
yum list	列出资源库中所有可以安装或更新以及已经安装的rpm包
yum list <regex>	列出资源库中与正则表达式匹配的可以安装或更新以及已经安装的rpm包
yum list available	列出资源库中所有可以安装的rpm包
yum list available <regex>	列出资源库中与正则表达式匹配的所有可以安装的rpm包
yum list updates	列出资源库中所有可以更新的rpm包
yum list updates <regex>	列出资源库中与正则表达式匹配的所有可以更新的rpm包
yum list installed	列出资源库中所有已经安装的rpm包
yum list installed <regex>	列出资源库中与正则表达式匹配的所有已经安装的rpm包
yum list extras	列出已经安装的但是不包含在资源库中的rpm包
yum list extras <regex>	列出与正则表达式匹配的已经安装的但是不包含在资源库中的rpm包
yum list recent	列出最近被添加到资源库中的软件包
yum search <regex>	检测所有可用的软件的名称、描述、概述和已列出的维护者，查找与正则表达式匹配的值
yum provides <regex>	检测软件包中包含的文件以及软件提供的功能，查找与正则表达式匹配的值
yum clean headers	清除缓存中的rpm头文件
yum clean packages	清除缓存中rpm包文件
yum clean all	清除缓存中的rpm头文件和包文件
yum deplist <packages>	显示软件包的依赖信息

- 当第一次使用yum或yum资源库有更新时，yum会自动下载所有所需的headers放置于 /var/cache/yum 目录下，所需时间可能较长。
- 还可以使用 yum info 命令列出包信息，yum info 可用的参数与 yum list 的相同。
- yum 命令还可以使用 -y 参数用于用 yes 回答命令运行时所提出的问题。

yum 命令工具使用举例

1、升级系统

```
#  yum update
```

2、安装指定的软件包

```
#  yum install w3m lync
```

3、升级指定的软件包

```
#  yum update w3m lync
```

4、卸载指定的软件包

```
#  yum remore w3m lync
```

5、查看系统中已经安装的和可用的软件组

```
#  yum grouplist
```

6、安装指定软件组中的软件包

```
#  yum groupinstall "Virtualization"
```

7、更新指定软件组中的软件包

```
#  yum groupupdate "Virtualization"
```

8、卸载指定软件组中的软件包

```
#  yum groupremore "Virtualization"
```

9、清除缓存中的rpm头文件和包文件

```
#  yum clean all
```

10、搜索相关的软件包

```
#  yum search python
```

11、显示指定软件包的信息

```
#  yum info python
```

12、查询指定软件包的依赖信息

```
#  yum  deplist  python
```

13、列出所有以 yum 开头的软件包

```
#  yum  list  yum\*
```

14、列出已经安装的但是不包含在资源库中的rpm包

```
#  yum list extras
```

如果您安装了并非来自仓库的软件，当它有新版时，yum update 无法自动更新它。为保证您总是用着最新的软件，可以订阅一份电子邮件或 RSS 服务，这样当有新版时可以得到通知。

加速 yum 的下载

使用全局代理

要设置所有 yum 操作都使用代理服务器，可以在 /etc/yum.conf 中设置代理服务器的信息。例如：

```
# 代理服务器:端口号
proxy=http://mycache.mydomain.com:3128
# 设置用于yum 连接的帐户细节
proxy_username=yum-user
proxy_password=password
```

为单个用户配置代理服务器

要为一个特定的用户启用代理访问，只要将实例框中的文本行加入这个用户的 shell 配置中。对于默认的 bash shell，配置是在 ~/.bash_profile 中。例如：

```
# 当前用户使用的匿名代理服务器
http_proxy="http://mycache.mydomain.com:3128"
export http_proxy
```

或

```
# 当前用户使用的代理服务器
http_proxy="http://yum-user:password@mycache.mydomain.com:3128"
export http_proxy
```

使用 yum 的扩展插件 yum-fastestmirror

当仓库配置文件（*.repo）使用 mirrorlist 或在 baseurl 中指定多个URL 时，yum-fastestmirror 可以自动选择最快的镜像站点。要想使用 yum-fastestmirror 加快下载速度，使用如下命令安装即可。

```
#  yum -y install yum-fastestmirror
```

yum-fastestmirror 的配置文件是 /etc/yum/pluginconf.d/fastestmirror.conf，一般无需修改。

系统自动更新

yum-updatesd

CentOS 系统默认安装了 yum-updatesd 软件包。

```
$ rpm -ql yum-updatesd
/etc/dbus-1/system.d/yum-updatesd.conf
/etc/rc.d/init.d/yum-updatesd
/etc/yum/yum-updatesd.conf
/usr/sbin/yum-updatesd
/usr/share/man/man5/yum-updatesd.conf.5.gz
/usr/share/man/man8/yum-updatesd.8.gz
```

由此可见，系统在启动过程中运行了 yum-updatesd 守护进程。这个守护进程读取 /etc/yum/yum-updatesd.conf 配置文件。

```
[main]
# 设置检查更新的时间间隔 (单位为秒)
run_interval = 3600
# 为了避免过于频繁的访问更新服务器设置检查更新的最小时间间隔 (单位为秒)
updaterefresh = 600

# 设置如何发送更新通知(有效值: dbus, email, syslog)
emit_via = dbus
# should we listen via dbus to give out update information/check for
# new updates
dbus_listener = yes

# 配置是否自动安装更新
do_update = no
# 配置是否自动下载更新
do_download = no
# 配置是否自动下载有依赖关系的更新
do_download_deps = no
```

yum-cron

还可以使用 yum-cron 自动安排 cron 任务。要使用 yum-cron 需要使用如下命令安装：

```
#  yum -y install yum-cron
```

使用如下命令可知 yum-cron 包安装的文件：

```
$ rpm -ql yum-cron
/etc/cron.daily/yum.cron            #  每日更新脚本
/etc/cron.weekly/yum.cron          #  每周更新脚本
/etc/rc.d/init.d/yum                    #  启动脚本
/etc/yum/yum-daily.yum
/usr/share/doc/yum-cron-0.1
/usr/share/doc/yum-cron-0.1/COPYING
/usr/share/doc/yum-cron-0.1/README
```

要激活自动更新，输入如下命令：

```
#  chkconfig --level 345 yum on
#  service yum start
```

参考

- 使用 yum 管理软件 [http://www.proxyserve.net/index.php?q=aHR0cDovL2RvY3MuaHVpaG9vLmNvbS95dW0vbWFuYWdpbmctc29mdHdhcmUtd2l0aC15dW0temhfY24vaW5kZXguaHRtbA%3D%3D]
- 关于Fedora Core 5.0 通过Yum在线升级说明 [http://www.proxyserve.net/index.php?q=aHR0cDovL2ZlZG9yYS5saW51eHNpci5vcmcvbWFpbi8%2FcT1ub2RlLzEzMg%3D%3D]
- Fedora/Redhat 在线安装更新软件包，yum 篇 [http://www.proxyserve.net/index.php?q=aHR0cDovL2ZlZG9yYS5saW51eHNpci5vcmcvbWFpbi8%2FcT1ub2RlLzMx]
- F7 Setup for Installing Applications — yum, etc. [http://www.proxyserve.net/index.php?q=aHR0cDovL29wdGljcy5jc3VmcmVzbm8uZWR1L35rcmllaG4vZmVkb3JhL2ZlZG9yYV9maWxlcy9mNy9ob3d0by95dW0uaHRtbA%3D%3D]
- 显示源文件
- 登录