

Deploying and Managing Linux Desktops With Kickstart and APT



An solution for deploying and maintaining multiple identical Linux desktops / workstations, using a combination of Red Hat's automated install method and the APT packaging tool.

The talk should be particularly of interest to people maintaining large groups of Linux machines, and most content is applicable to more than one Linux distribution. Additionally, Red Hat users who have yet to experience the joys of apt-get will discover the wonders of this rather nifty tool to assist the installation of RPM packages on their systems.

What Makes a Good Workstation?

- ◆ Allows access to resources staff need to get their job done
 - ◆ Software
 - ◆ NFS shares (staff home directories, as well as file / application servers)
 - ◆ Accounts (via LDAP, NIS, SMB or another system)
 - ◆ Printers
 - ◆ Networking and other settings
 - ◆ Profile variables (eg, proxy settings)
 - ◆ Backup settings
 - ◆ Legal disclaimers
 - ◆ Other site specific things (/History, site specific cron jobs)
- ◆ Has working software (up to date with security and bugfix releases)
- ◆ Is easy to administer and upgrade
 - ◆ **Works like any other standard system**
 - ◆ Packaged apps wherever possible (generally, its **always** possible) with a little time and knowledge - and APT lowers these barriers further
 - ◆ Unpackaged apps installed in a separate, correct location (generally /usr/local)

Why Standardize on a Deploy / Maintenance System?

- ◆ Time savings when deploying machines - no user interaction required
- ◆ Known configuration assists troubleshooting (process of elimination allows one to easily determine whether a problem is software or hardware soecific)
- ◆ Staff portability between desktops
- ◆ Makes it harder to `miss the gaps' when configuring a machine
- ◆ Can enforce policies about machine use, appropriate software (more on that later)
- ◆ Encourage good system administrator practices

Status

- ◆ Kickstart system has been used for sixteen installs, most of which use Red Hat 7.2 version
- ◆ Apt being used on a similar number of machines
- ◆ Took around twenty hours to get right, most of the time spent waiting for installs
- ◆ Will soon move all remaining Cyber desktop's to use the system
- ◆ Might be part of maintaining servers and client boxes

A Little About RPM And APT

- ◆ RPM is a packaging system for Linux.
 - ◆ It is capable of packaging both source code and binaries.
 - ◆ The original sources are combined with a spec file which provides information about the software contained within (version numbers, other necessary software, incompatible software, compilation procedures, installation procedures, patching procedures, upgrade procedures, etc)
 - ◆ By `rebuilding' the SPEC file, binary and source packages are created
 - ◆ RPM 3 is the current Linux standard, this will hopefully be updated to 4 as soon as the book comes out
- ◆ APT is a tool which runs on top of a packaging system (like dpkg and rpm)
 - ◆ Basically a front end to other tools
 - ◆ Can fetch software over a variety of means (http, cdrom, nfs, etc) and grab any dependencies, and install the software
 - ◆ Uses the same mechanisms as the underlying package system to install software (in the RPM version, librpm and the rpm binary).
 - ◆ Does an **excellent** job of preserving system integrity, making sure the system never gets into an unreliable state.

About Kickstart

- ◆ A way of automating the answers to a Red Hat install
- ◆ Works over CDROM, http, NFS, and sometimes SMB
- ◆ Allows one to tune
 - ◆ What packages are installed
 - ◆ Networking information
 - ◆ Bootloader
 - ◆ Partitioning
 - ◆ Firewalling
 - ◆ Root password and general authentication (LDAP, Radius, etc)
- ◆ Provides a post install script to allow further configuration

Installing the Machines

- ◆ Using Red Hat kickstart over HTTP
- ◆ Post install script does a few post setup tweaks (we're generally replacing these with custom packages)
- ◆ Post install script installs apt (rpm -Uvh <http://apt/apt-version.rpm>)
- ◆ Post install script configures APT sources
- ◆ Apt uses rpm and rpmlib to fetch and install additional software, and whatever other apps they rely upon.
- ◆ Currently, this is around 250 additional packages, and install process takes around an hour depending on network load

A Kickstart File

- ◆ Over to editor

An APT Server

- ◆ DocumentRoot
 - ◆ redhat-7.2
 - ◆ i386
 - ◆ Base
 - ◆ RPMS.main
 - ◆ RPMS.updates
 - ◆ RPMS.kde
 - ◆ RPMS.gnomehide
 - ◆ SRPMS.main
 - ◆ SRPMS.updates
 - ◆ SRPMS.kde
 - ◆ SRPMS.gnomehide

In your sources.list

```
rpm http://webservice redhat-7.2/i386 main updates kde gnomehide  
rpm-src http://webservice redhat-7.2/i386 main updates kde gnomehide
```

Maintaining the Machines

- ◆ Wanted to be able to have changes be pulled in by desktops, rather than having to make-changes
- ◆ Using cron, apt and rpm solves this problem neatly.
- ◆ We create a package. Its dependencies, obsoletes, and version number describe the configuration of a cybersource-workstation at that point in time
- ◆ Cron uses apt to install the latest package each night

Some practical examples

- ◆ We wanted the Fluxbox desktop to be installed on each machine. Update the cybersource-workstation package spec to require fluxbox, and `rebuild' the package
- ◆ Some particular packages were taking up disk space and emailing root accounts as they hadn't been set up. We made each machine remove these packages by making a newer version of the cybersource-workstation `obsolete' them
- ◆ Many custom configurations are stored as packages. A Sendmail-localserver package allows staff to send email from applications running on their workstations with a correct return address, using our offices central mail server as a `hub'. This uses sendmail's `nullclient' option.
- ◆ Printer config packages

Maintaining the Repository

- ◆ Staff drop new packages in a directory (linked away from the heirarchy of the apt repository)
- ◆ A script regenerates the index files for the directory.
- ◆ Documentation + answers to FAQs is default page for respository
- ◆ Kickstart changes are in CVS
- ◆ APT changes are not under version control, yet.

Benefits

- ◆ Easily deploy/ maintain new workstations
- ◆ Staff more portable
- ◆ Staff easily know if packages are available
- ◆ Less crap (unpackaged apps) installed on desktop
- ◆ We have 3000 packages in our repository. Whatever it is, its probably there.
- ◆ More `uncommon' apps installed by packages - eg, bind-chroot, postfix, JRE 1.4,
- ◆ Staff can update remote machines via the magic of ssh

Problems

- ◆ Lack of responsiveness for long kickstart install scripts, which occur in the background. I'd like to be able to interact with the installer using a dialog like program during kickstart installs.
- ◆ Red Hat don't test Kickstart as much as the rest of their distro
- ◆ Bad packages. These can generally be traced to one particular source of packages.
- ◆ SSH port forwarding with Apache virtual servers
- ◆ I'd love to make this publically available but I do not have the bandwidth

Further Steps

- ◆ Sign repository
- ◆ Use a version control system for changes made to the repositories
- ◆ Automated updating of archives based on change time for directories
- ◆ Automated rebuild of source packages
- ◆ Building more profile / group packages

Thankyou!

Questions anyone?