

RED HAT :: SAN DIEGO :: 2007

SUMMIT



Red Hat Enterprise Linux 5, Yum and You

Jeremy Katz

Previous RHEL Releases

- Various software management tools completely separate
 - anaconda
 - system-config-packages
 - up2date
- Different behavior depending on tool
- Bugfixes/enhancements had to be applied to multiple code bases



RHEL5

- All three tools use the same core backend code -- yum
- Consistency in behavior across tools
- Consistency in interface across tools
- One code base for bug fixes and enhancements



Transition

- Biggest and most noticeable difference is no more `/usr/bin/up2date`
 - Replaced with `/usr/bin/yum` (CLI) and `/usr/bin/pup` (GUI)
 - Command line arguments differ but are pretty straight-forward
- Installs/kickstart continue to work as always
- RHN web UI unchanged
- `system-config-packages` replaced with `pirut`
- `rhn-applet` replaced with `yum-updatesd` + `puplet`



New Functionality

- Multiple repository support across the software management stack
 - Repositories can be generated with the createrepo tool
- Ability to access site-specific software at install-time via kickstart
 - `repo --name=myrepo --baseurl=http://my.local.site/path/to/repo`
- Post-install package installation tool (pirut) aware of updated packages
- Plugins allow for extending functionality
- API for writing your own tools as well



New Functionality via Plugins

- Rich plugin system with the hooks for many types of new functionality
 - Implement site-specific policies (installonlyn, versionlock)
 - Add new modes (downloadonly)
 - New criteria for how to select updates (yum-security, skip-broken)
 - New functionality to allow for interactive config change merging (mergeconf)
- Hooks exist to call out at 13 different points
- Differentiation between interactive (yum-cli) and core plugins



Simple Tools Built on Yum API

- API is rich and simple to allow for easily building your own tools on top of the base
- Tools can use everything that plugins provide
- yum-utils has a set of ready to go utilities (repoquery, yumdownloader, yum-builddep)
- Simplest tool is something like (yum 3.2 simplifications)

```
import yum, sys
```

```
for g in yum.YumBase().comps.groups:
```

```
    if sys.argv[1] in g.packages:
```

```
        print g.groupid
```



Evolving and Community Driven

- Yum has been used within Fedora for several years now
- Active community project with a vibrant upstream
 - Several heavily active committers to the core with other more occasional committers
 - More contributors via plugins and simple utilities in yum-utils
- Yum 3.2 brings more new features as driven by users
- API cleanups continue to make the API even easier to use while doing as much as possible to retain compatibility
- New functionality starts as a plugin/separate utility, then moves into yum core as it matures
- Multiple frontends to meet any need



More Information

- yum man pages
- <http://linux.duke.edu/projects/yum/>
- <http://lists.linux.duke.edu/mailman/listinfo/yum>
- <http://lists.linux.duke.edu/mailman/listinfo/yum-devel>

