

RED HAT :: SAN DIEGO :: 2007

SUMMIT



Handling Driver Updates in RHEL5 and Beyond

Jon Masters

May 18th, 2007

What are “Driver Updates”?

- The RHEL kernel is based upon the “upstream” Linux kernel from kernel.org.
 - But with various fixes and enhancements as necessary for customers.
- Third parties want to add their own drivers to RHEL.
 - Sometimes these drivers are shipped in a later update release.
 - Sometimes the driver is needed yesterday, not in 6 months.
 - Sometimes the driver supplier won't send driver upstream.
 - Red Hat maintain a stable kernel ABI to assist with this.
- Driver Updates are Linux device drivers (and other kernel modules) that add some kind of third party functionality to the RHEL kernel at runtime.



Overview

Today's presentation is split into the following parts...

- Driver Updates in the past
 - What problem do we want to solve?
 - Why is this important?
- Driver Updates in RHEL5
 - What do we have right now?
 - How does it work?
- Driver Updates in the future
 - Where do we want to be in a year?
 - What's the longer term outlook?



Driver Updates in the past...

- No standardize mechanism for adding drivers to a system
 - Clumsy, manual builds, required on every kernel update.
 - Customers don't want to have to rebuild drivers.
 - Everyone had their own (different) process.
 - Didn't scale well.



Driver Updates in the past...

- Typical process for deploying a driver update:
 - Download the driver source code from third party website.
 - Maybe a pretty GUI installer is available
 - But probably not. Different every time.
 - Ensure kernel-devel package, GCC toolchain and other deps. in place.
 - `$ yum install kernel-devel`
 - Build driver update using tools on target machine.
 - `$ make -C /lib/modules/`uname -r` /source modules`
`M=`pwd``
 - Install the driver into the current kernel.
 - `$ make -C /lib/modules/`uname -r` /source`
`modules_install M=`pwd``
 - Repeat every time a new kernel is released.



Driver Updates in the past...

- Repeat every time a new kernel is released.
 - On security updates.
 - On kernel errata.
 - On minor upgrades.
 - Every single target.
 - Every single time.
 - Does NOT scale.
- Even if the driver were compatible with the new kernel
 - Module is unchanged.
 - Module versioning (modversions).
 - ABI stability in RHEL kernels.



Driver Updates in the past...

- Other issues with using third party Driver Updates could include:
 - Requires customers to understand kernel build processes.
 - Not every sysadmin is also a kernel engineer for fun.
 - Not always a sysadmin available (e.g. Desktop).
 - Not in a clean pre-built binary RPM package.
 - Requires build/devel tools installed on production machines.
 - Hard to ensure consistent build process on target.
 - Many customers want to avoid doing this.
 - Does NOT scale.



Driver Updates in the past...

- Third parties attempted to solve the problems for themselves.
 - Many different solutions to the same problem.
- Need a standard solution for everyone to use.
 - Easy to understand, deploy and use.
 - Must scale to many systems.
 - Must be supportable.



Driver Updates in RHEL5

- Driver Updates are regular RPM packages. Install them using whatever RPM tools are already in use in a given Red Hat Enterprise Linux 5 environment.
- Typical process for deploying a driver update:
 - `$ rpm -ivh kmod-ipw3945-1.0-4.7.el5.i686.rpm`



Driver Updates in RHEL5

- RHEL5 tracks third party driver compatibility in the distribution.
 - Drivers work across kernel updates and RHEL5 updates.
 - Customers don't need to know how to build drivers.
 - Customers can use RHN Satellites, Yum repos...
 - Scales much better.



Driver Updates in the RHEL5

- RHEL5 makes modifications to the kernel packaging process.
- Add kernel symbol version tracking to its RPM package.
 - `$ rpm -q --provides kernel`
 - `kernel(rhel5_block_ga) = a42055e630b73ddd7254fc6963814e16913852e5`
 - `kernel(rhel5_fs_ga) = b96eba087460900b3aa6064930cf23d58908d4d6`
 - `kernel(rhel5_kernel_ga) = 2cd142708e2d573b2de522df5df87aaeb7c1d298`
 - `kernel(rhel5_kernel_irq_ga) = 828c3f468e7a640d409d2bf24b4676e457406917`
 - `kernel(rhel5_kernel_module_ga) = 1b051ce57d6b18fdf071786f6f7296d3d0ab28f9`
 - `kernel(rhel5_kernel_power_ga) = 3c2c37d553ebecf99e6d147387f3dd4b5f5df7b7`
 - `kernel(rhel5_lib_ga) = 088a6b77cde4f82c65b0d7f34802cfa41d209328`
 - `kernel(rhel5_mm_ga) = 09f63dfab81bba7e01a2bf693f5ce125db466051`
 - `<etc>`



Driver Updates in the RHEL5

- RHEL5 makes modifications to the module (driver) packaging process.
- Drivers track kernel symbol requirements via RPM dependencies.
 - `$ rpm -q --requires kmod-ipw3945`
 - `kernel(rhel5_drivers_base_ga) = 2490efa8790e7d4b3c7ae4536866c48f1334a356`
 - `kernel(rhel5_drivers_char_ga) = 9825f97c6773e4d766359d55afa360b0c31cd1a6`
 - `kernel(rhel5_drivers_pci_ga) = 33d9e0612417f727a0d8311c229fe8a80c65bb65`
 - `kernel(rhel5_fs_sysfs_ga) = bb43e88ba2b9ebed086513342d9a5a18ed9355a7`
 - `kernel(rhel5_kernel_ga) = 2cd142708e2d573b2de522df5df87aaeb7c1d298`
 - `kernel(rhel5_kernel_irq_ga) = 828c3f468e7a640d409d2bf24b4676e457406917`
 - `kernel(rhel5_kernel_module_ga) = 1b051ce57d6b18fdf071786f6f7296d3d0ab28f9`
 - `kernel(rhel5_lib_ga) = 088a6b77cde4f82c65b0d7f34802cfa41d209328`
 - `kernel(rhel5_mm_ga) = 09f63dfab81bba7e01a2bf693f5ce125db466051`
 - `<etc>`



Driver Updates in the RHEL5

- RHEL5 groups kernel ABI symbol versioning information (for ease of management and to alleviate huge dependencies).
 - `$ tar xvfz /usr/src/kernels/2.6.18-8.1.2.el5-i686/symsets-2.6.18-8.1.2.el5.tar.gz`
 - `$ less symsets-2.6.18-8.1.2.el5/rhel5_kernel_ga.2cd142708e2d573b2de522df5df87aaeb7c1d298`
 - `0x7e0221e4 complete kernel/built-in.o`
 - `0xda4008e6 cond_resched kernel/built-in.o`
 - `0xdf92b38 copy_fs_struct kernel/built-in.o`
 - `0x4f3f1813 current_fs_time kernel/built-in.o`
 - `0xdc43a9c8 daemonize kernel/built-in.o`
 - `0xffd5a395 default_wake_function kernel/built-in.o`
 - `0x4827a016 del_timer kernel/built-in.o`
 - `0x72e90a52 dequeue_signal kernel/built-in.o`
 - `0x0b1ddd1b destroy_workqueue kernel/built-in.o`
 - `0x7ea7d551 do_exit kernel/built-in.o`
 - `0x72270e35 do_gettimeofday kernel/built-in.o`
 - `0xf0a529b7 do_softirq arch/i386/kernel/built-in.o`
 - `<etc>`



Driver Updates in the RHEL5

- RHEL5 adds scripts to handle kernel upgrades transparently.
 - Modules installed in `/lib/modules/$kernel/extra/$name/$module.ko`
 - `/sbin/weak-modules` runs on kernel install or/and upgrade.
 - Ensures that symlinks are created for compatible kernels.
 - Ensures that initrd images are updated correctly.
 - Ensures that module loading deps. are updated.
- RHEL5 does not officially support overriding supplied drivers
 - Driver Updates intended only to add new drivers.



Driver Updates in the RHEL5

- Installing an example RHEL5 Driver Update RPM kmod module package.
 - Locate the module on the RHEL5 supp. CD.
 - `$ rpm -ivh kmod-ipw3945-1.0-4.7.el5.i686.rpm`
 - IPW3945 Driver Update is installed.
 - `/lib/modules/2.6.18-8.el5/extra/ipw3945/ipw3945.ko`
 - Links automatically added for compatible kernels.
 - `/lib/modules/2.6.18-8.1.2.el5/weak-updates/ipw3945/ipw3945.ko -> /lib/modules/2.6.18-8.el5/extra/ipw3945/ipw3945.ko`
 - Dependent files updated, autoload next boot.
 - Or `modprobe` module in post-install.



Driver Updates in the RHEL5

- Example of packaging a RHEL5 Driver Update kmod module.
 - Locate the module SRPM (RHEL5 Supp. CD.).
 - `$ rpm -ivh ipw3945-kmod-1.0-4.7.el5.src.rpm`
 - `$ rpmbuild -ba --target i686 /usr/src/redhat/SPECS/ipw3945.spec`
(change target to match your build environment)
 - Standard kernel module build process will run.
 - Notice the RPM build output.
 - Requires: kernel(rhel5_net_ieeee80211_ga) = 7ead0ef2c63860e95456e3dffe41afdd94b16661 kernel(rhel5_mm_ga) = 5016f99aaf599aa846aa6937d9f20b5b1c4e3575 kernel(rhel5_net_sched_ga) = f37da797205bc94ee3ef5c164fe5e1f648db3483 kernel(rhel5_kernel_ga) = 52c756e7a0505cfbcd0285fd99a855956b18c37b...
 - RPM binary packages in /usr/src/redhat/RPMS
 - /usr/src/redhat/RPMS/i686/kmod-ipw3945-1.0-4.7.i686.rpm



Driver Updates in the RHEL5

- RHEL5 Driver Updates must be built for each pertinent kernel variant.
 - 10 RHEL5 kernel variants in GA (RHEL5.0.).
 - i686, i686-PAE, i686-xen
 - ia64, ia64-xen
 - ppc64, ppc64-kdump
 - s390x
 - x86_64, x86_64-xen
 - But only those you need to ship actual RHEL5 Driver Updates against.
 - No practical need for an s390x laptop with IPW3945.



Driver Updates in the RHEL5

- RHEL5 has a standardized Driver Update RPM SPEC file format.
 - Every RPM is built using a SPEC file, defining build process.
 - Calls to kmodtool abstract away boilerplate detail.
 - redhat-rpm-config (/usr/lib/rpm/redhat) updated.
 - Various scripts automate post-processing
 - Add kABI tracking deps. automatically.
 - Verify against the RHEL5 kernel ABI.
 - Uses RHEL5 ABI symbol whitelist.
 - Further enhancements planned.
 - Examples at <http://www.kerneldrivers.org/>



Driver Updates in the RHEL5

- RHEL5 makes many improvements over previous Linux driver situation.
 - Introduces stable RHEL5 ABI symbol whitelist for driver packagers.
 - Additional tools to validate whether drivers use official symbols.
 - Introduces RPM based RHEL5 driver packaging infrastructure.
 - Handle kernel upgrades without breaking existing drivers.
- But there is more work still to be done on driver support.
 - Automate driver detection/location.
 - Graphical management tools.
 - Transparency.
 - Scalability.



Driver Updates in the future...

- Current Driver Updates mechanism is an improvement over rebuilding modules.
 - Customers don't need to use devel tools.
 - But still manual driver installation.
 - Need to know where drivers are.
 - Need to manage driver updates.
- We really need to automate the entire device detection and installation process.
 - Provide the user with a GUI tool for hardware device management.
 - Automatically determine the correct driver for any given device.
 - Security policies and ability to define driver vendor preference.
 - Download, install, and update the driver whenever needed.
 - Make driver management as seamless as possible.



DriverTool (aka Project Awesome)

- Future driver management will utilize an online service.
 - Locate available drivers for hardware by using modalias.
 - XMLRPC based (online service).
 - Support different distributions and packaging.
 - Support build from source if desired.
- GUI extensions to hal-device-manager.
 - Automatically notify the user when a driver is available.
 - Provide a list of alternatives.
 - Support upgrading drivers.



DriverTool (aka Project Awesome)

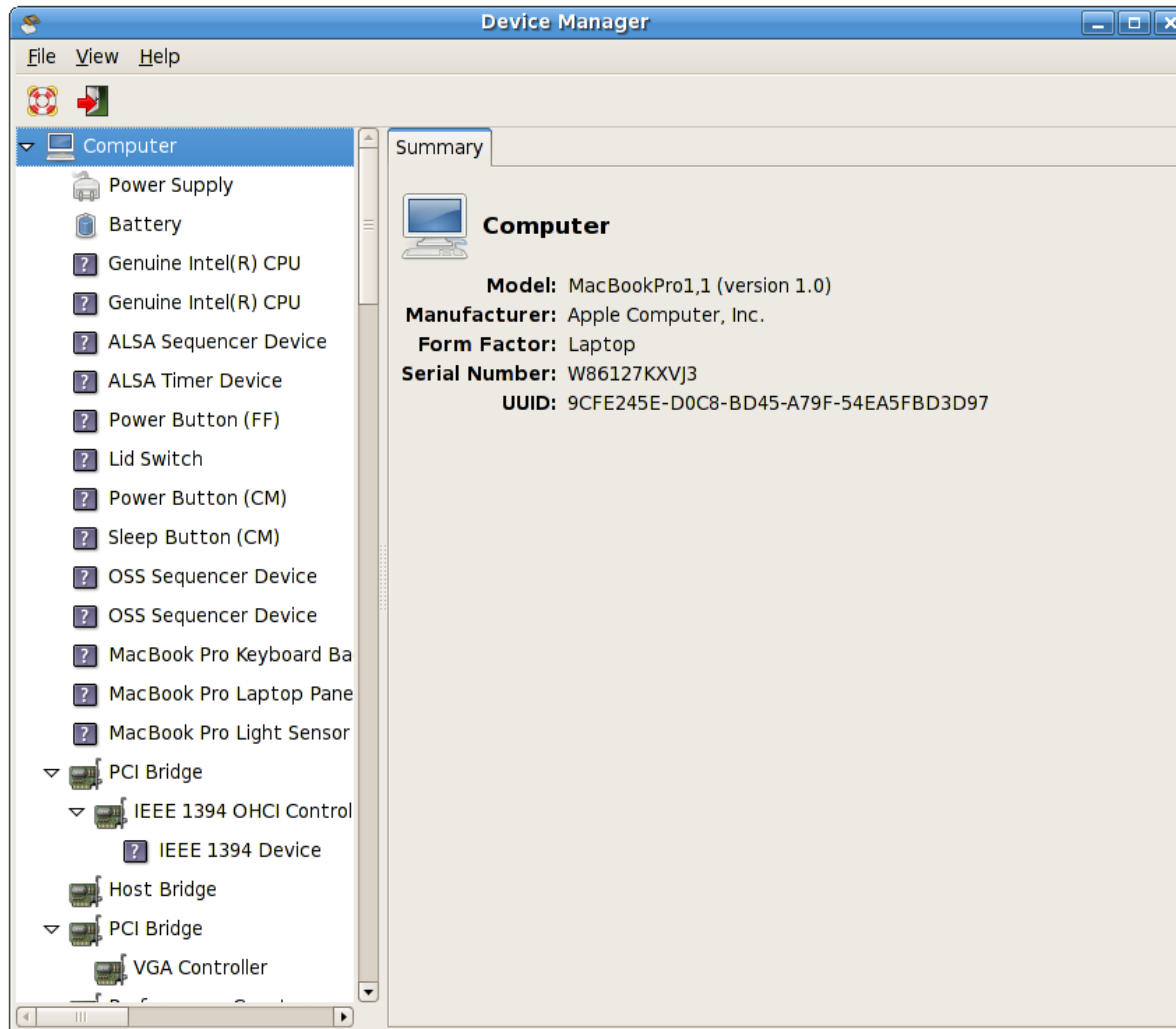
- Example query to online XMLRPC service:

```
[jcm@perihelion drivertool]$ ./dt_quicktest "pci:v00008086d00004222sv*sd*bc*sc*i*"
standard_version: 2007033100.
id: 1.
name: ipw3945.
version: 1.1.0d.
type: Linux Driver Module.
vendor: Intel Corporation.
package author: Jon Masters <jcm@redhat.com>.
package vendor: Red Hat, Inc..
description: Intel(R) PRO/Wireless 3945 Network Connection Driver for Linux.
alias: pci:v00008086d00004227sv*sd*bc*sc*i*.
alias: pci:v00008086d00004222sv*sd*bc*sc*i*.
os_name: Red Hat Enterprise Linux.
os_version: 5.0.
os_target: i686.
type: RPM.
url: http://localhost/drivertool/ipw3945/kmod-ipw3945-1.0-4.7.el5.i686.rpm.
```



DriverTool (aka Project Awesome)

- Hal-device-manager graphical device management tool already in GNOME.



DriverTool (aka Project Awesome)

- Python-based build/install framework behind the scenes for driver provisioning.
 - Do for driver updates what NetworkManager does for WiFi.
 - Request driver availability via drivertool service.
 - Download driver or source from remote site.
 - Rebuild/install driver as required.
 - Driver rebinding support TBD.
 - Handle kernel upgrades.
- What we have implemented of this plan so far.
 - online XMLRPC service.
 - Simple client tools/framework.



Driver Updates longer term...

- Ideally all drivers would be upstream in the official Linux kernel.
 - We want to help this process and not to hinder it.
 - But sometimes driver updates are needed.
 - Need to strike a balance.
 - Respect the community.
- Linux users want simple, graphical, driver management using GUI tools.
 - They don't want to know about kernel build processes.
 - They want to be able to see which drivers are in use.
 - They want to handle system upgrades automatically.
 - RHEL5 begins the process of making this easier.



Questions...

- Please feel free to ask your questions in the time we have left.
 - Or send email to jcm@redhat.com

- This presentation contains information related to ongoing R&D for potential use in future Red Hat products. It should not be taken as product commitment.

