Northwestern University Dell Kace Patch Management

Desktop Patch Management Best Practices

Table of Contents:

Audience:

Northwestern University employees and affiliates involved with maintaining desktop and laptop client workstations. This document focuses on the use of Dell Kace K1000 for patching however the same common principles and best practices can be applied to other client management solutions.

This document is intended to only be a high level overview of client patch management as it relates to Dell Kace and is comprised of information gleaned from other sources and experience in patch management. For individuals who are new to patch management, please see some of the documents listed in the Appendices. In addition, most large commercial vendors will also publish patching guides for their products that can provide valuable insight into patch management.

Definition:

The goal of patch management is to mitigate the risk to the integrity of end-user workstations and to the network as a whole providing a secure, functional, and reliable product. This document outlines key best practices one should consider implementing when managing the patch level of computers, as well as general instructions for creating specific patch schedules

using the Dell Kace K1000 solution.

Patch Management Approaches:

When implementing a patch management policy within your area there are three basic approaches: “Fire & Forget”, “Fire & Forget (Delayed)”, and “Review-Test-Deploy”. Each IT group will need to decide which patch management approach works best for them based on end user computing needs, local security policies, and available resources.

Fire & Forget Method

Configuring an automated patch schedule and then letting the patch management solution deploy patches to workstations as they become available. In this model, patches are neither staged, nor tested internally, prior to being released to the target general computing population. This approach does not attempt to limit the possibility of deploying a problematic patch.

Fire & Forget (Delayed) Method

Same basic approach as above, however, restricting deployable patches to only those items that have been released for a predefined period of time. (i.e. 5 days, 7 days) This method does limit some potential for deploying a problematic patch by allowing time for the solution to supersede recently released patches, however no local verification or testing is completed prior to releasing to the target general computing population.

Review, Test, Deploy Method

This approach requires an established testing process. Patch testing is conducted by releasing patches to a test client control group prior to general computing population. If no issues are reported from the control group, patches are deployed to general computing population at a regularly scheduled interval. This approach is the best method for handling patch management within a school or department however takes additional resources and time to setup and maintain.

Guideline for Review, Test, Deploy Patch Management:

Preparation

1. A good patch management strategy requires the dedication of staff to assume the role of patch management technology owner. The amount of time greatly varies depending on the size of the department/school, the complexity of the endpoint environment, and the patch management strategy employed (which will be defined later in this document). A department/school should have at least two technology owners who oversee the patch status of client workstations and understand machine configurations well enough to perform a proper risk assessment of patches. After such assessment, the technology owners will schedule when the patches will be deployed.

2. Technology owners need to monitor listservs and websites for patch release announcements. All major software vendors[1] have some announcement mechanism for alerting individuals to the availability of new patches. The Dell Kace - K1000 publishes a daily email of new patches. In addition, there are third party websites and lists dedicated to the dissemination of patching related information. (Examples: SEC_UNITS@listserv.it.northwestern.edu, http://www.patchmanagement.org/, and lyris@listserv.patchmanagement.org)

3. Consider deploying patches on a regular interval (weekly, monthly, quarterly, etc.), regardless of if a risk assessment determined patches were necessary.

4. If an existing set of test scripts or testing procedures does not exist, they should be created to ensure that specific patches do not negatively impact critical business applications. An example of a test script would be a documented process of logging in to a critical application (e.g. NUFinancials) and navigating around to ensure functionality is maintained after the OS patch is applied.

5. Deploying patches to pilot patch testing control groups first is highly recommended.

6. Build a workflow, even an informal one, to deal with the process of reviewing patches, testing patches, and deploying patches. This can aid in allocating the proper amount of staff time to the patch management process.

Reviewing Vulnerabilities

1. Perform a risk assessment of new patches by reading and understanding release notes of the patch. This will aid in determining if a specific patch is needed on an endpoint and how quickly it should be installed. Examples:
a. Patches for web browsers do not need to be installed on servers where there is a more than reasonable expectation that no one will be browsing the Internet on the server. Conversely, users of desktops and laptops are almost always expected to browse the Internet, making patching web browsers and related plug-ins a necessity.
b. Patches that fix a critical zero-day vulnerability on an externally facing web server should be viewed as a top priority.
c. The risk assessment should also take into consideration the state of the environment. For example, computers dedicated to specific activities, such as monitoring scientific equipment, may not need to be patched as urgently due to the specialized nature of the work. Incompatibilities between newer versions of applications and critical line of business applications must be accounted for. See Kronos and Java.
2. When in doubt, install the patch.

Communication

1. Notify customers of planned updates if an outage or change in functionality is expected. Ideally customers should be given a minimum 24-hours' notice before the patch installation, but

the nature of the patch needs to be considered.
a. Critical security patches for a vulnerability which is being actively exploited, often known as a “zero-day” exploit, should probably not use the 24-hour rule and instead, a shorter notification window should be used.
b. Updates that introduce major functionality or appearance changes may need considerably more warning time.
2. Be consistent and concise when notifying customers of any outages and/or changes.
3. Communication can take the form of creating an established maintenance window where patches to their computers may happen. During this time they should be prepared for such work by ensuring all work is saved and they are logged off (if possible).

Install Patches

1. Installation of patches ideally should not occur during the business day or during peak usage but rather during off-peak times. For laptops and other types of computers which may not be constantly powered on, adjustments to the timing may be required.
a. Many patches require a system reboot after installation. It is generally a bad idea to force a system reboot during business hours or when a system is in use. Generally speaking, single user desktop computers can be rebooted during overnight hours (with a warning prompt before reboot). Laptops and computers connected to scientific equipment usually cannot be forced to reboot due to the risk of lost data and productivity.
2. Install patches in a test environment before deploying to production computers.
a. Test operating system and other “high level” patches on a clean computer first to rule out if any installation failures are due to compatibility issues with additional applications and not due to an issue with the patch itself.
b. Install patches on a test system that is configured as closely as possible to the production version and install patches as closely as possible to how they would be installed in production.
c. Use virtual machines with good snapshotting capabilities when testing patches, especially when testing patches on clean systems.

Post-Install

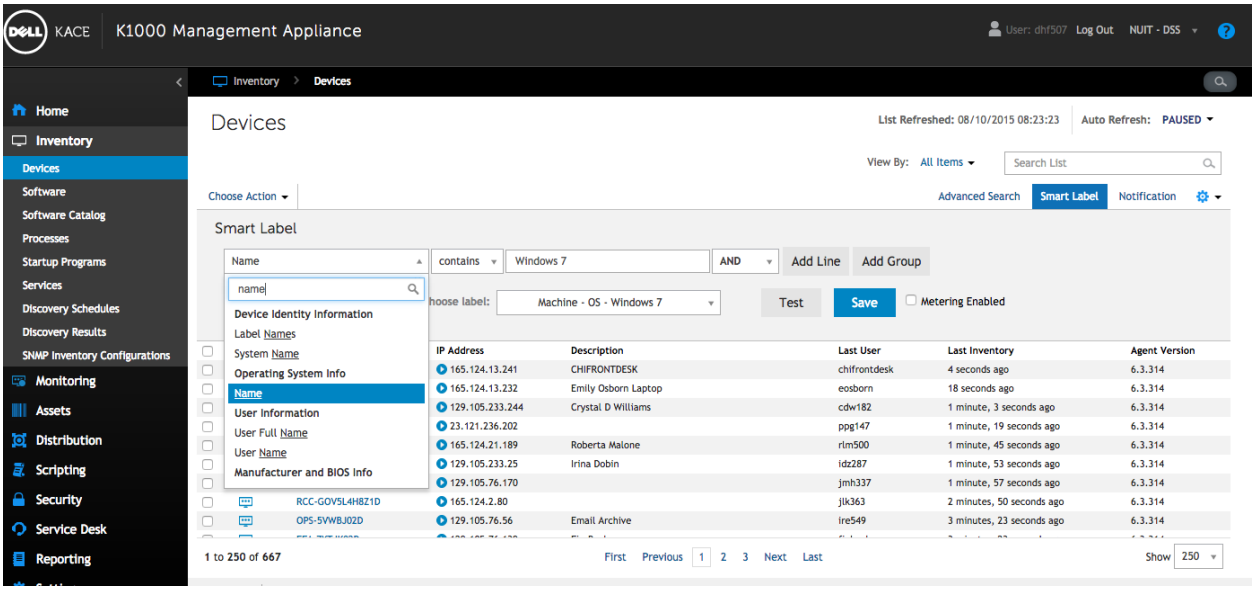
1. Use the reporting functionality of Dell Kace to determine if the deployment process worked as expected.
2. Have an established communication path (e.g. email, ticket) for end users to report possible patch problems. This is especially key for pilot groups.
3. Continue monitoring communication channels for possible issues with patches even after deployment.

Dell Kace Patch Configurations:

Machine Labels

To apply patches to a specific group of machines, Kace utilizes the same system of labels used for inventory, managed installations, scripts, and reports. Selection of machines for labels to be included in patch schedules should be based on some logical criteria such as:

- On campus versus off campus clients, for concerns related to bandwidth
- Laptop versus desktop clients - laptops can be on and off network, with a likelihood to be off overnight
- Machine role: computer lab, research lab, staff member, faculty member. It might be acceptable to force reboot an entire computer lab every night, but this is not a likely option for faculty and staff members
- Test machine control group
- Machines storing PII or PHI information



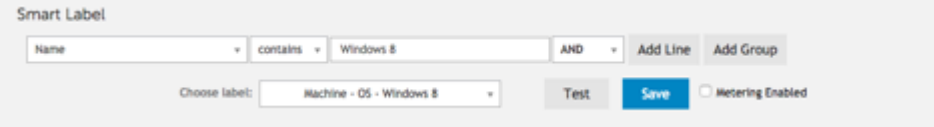
An example of a Smart Label that contains machines running Windows 7

Labels can be found under **Home -> Dashboard -> Label Management**. Generally, use of a “smart label” will provide the best results for including the proper clients to be targeted. A smart label will include all clients that meet the conditions that are defined in the label, rather than a manual label that is populated by manually searching for, and selecting machines for that label.

To create the machine smart label to be used for identifying clients to be patched, navigate to **Home -> Dashboard -> Label Management -> Smart Labels**, and click the “**Choose Action**” menu. Next, click New **-> Device Smart Label**. Now the conditions can be set to create the label for specific clients requiring patching.

For example, if you wish to only patch Windows 7 machines, but not Windows 8 machines,

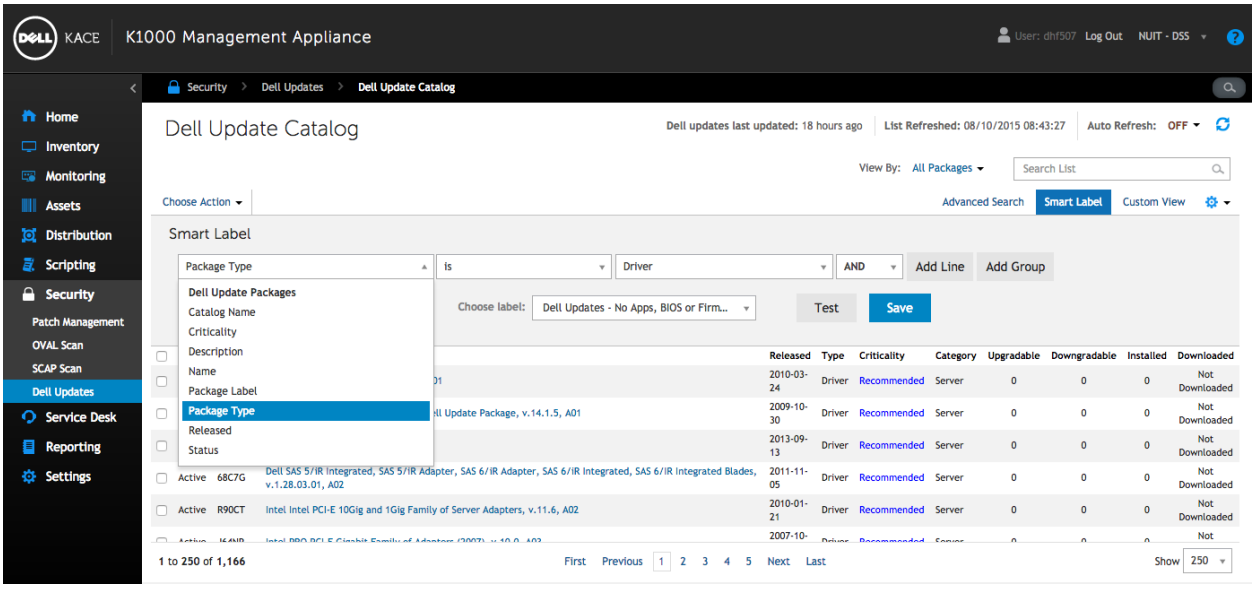
there are two routes to take. You could select the “**Operating System – Name**” field, select “does not contain” operator, and type “**Windows 8**” in the right-most field. Another option is to create a stand alone label that just includes Windows 8 machines, or any other machines that you do not want to receive patches, and then in this machine label, select the “**Label Names**” field, the “**!=**” operator, and the name of the label of the machines that we do not want to receive patching in the right-most field. When creating a smart label, there are many criteria for filtering, so look through all of them and you should find exactly what you are looking for targeting the correct machines for patching.



Detail: Creating a K1000 Machine Smart Label for Windows 8

Patch Labels

Patch labels group types of software patches so that they can be assigned to a schedule and group of machines. A patch label can be created one of two ways: either by creating a patch smart label, or a manual patch label. Selection of patches to include in labels should be based on some logical criteria such as patch severity, software title, release date, etc.

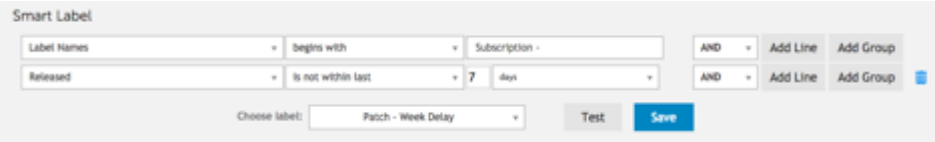


An example of a Patch Smart Label that only includes Dell driver updates

A patch smart label is created like any other smart label in the K1000. Navigate to **Home -> Dashboard -> Label Management -> Smart Labels**, and click the “**Choose Action**” menu. Choose **New -> Patch Smart Label**. An interface that looks very similar to the Machine Smart Label interface will appear, but the criteria are specific to patching.

If one would like to create a manual patch label, navigate to **Home -> Label Management ->**

Labels, and click the “**Choose Action**” menu. Choose **New Manual Label.** Fill in all required fields, and in the “Restrict Label Usage To” section, check the “**Patches**” box and click “**Save**.”

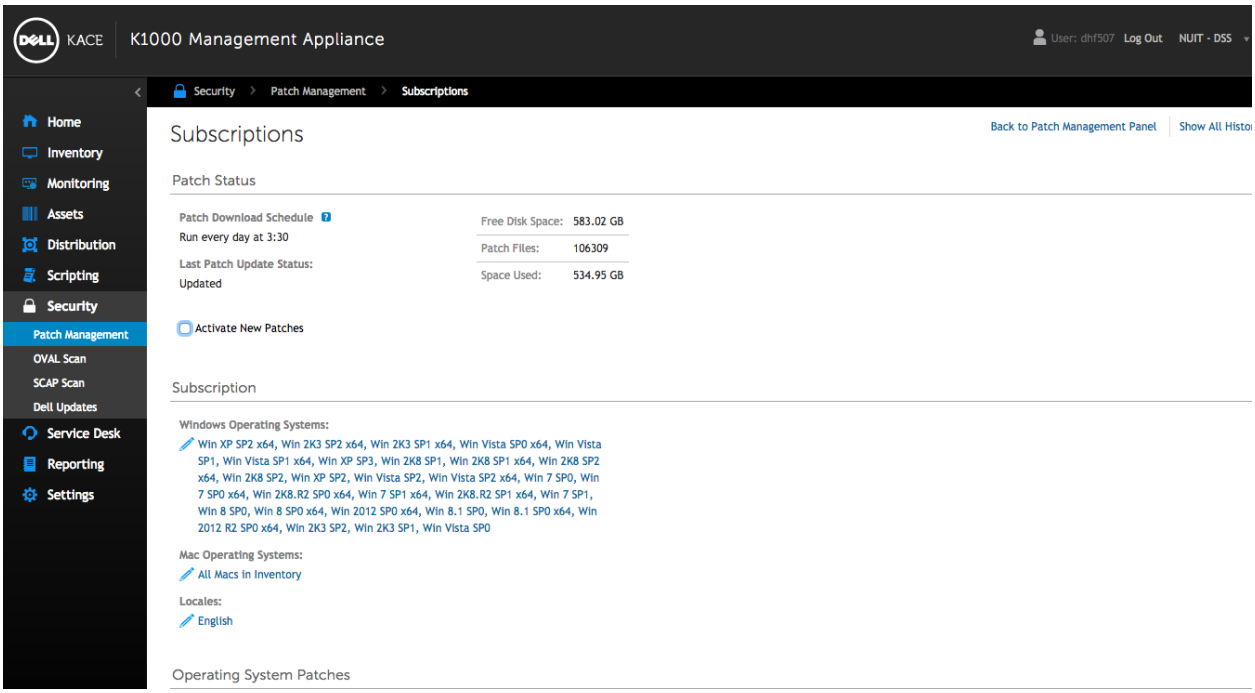


Detail: Setting up a K1000 Patch Smart Label by Release date

This label is now available for patches to be manually added via the patch catalog screen. This can be found in **Security -> Patch Management -> Catalog**. Using the checkboxes next to specific patches, you can select a particular patch and click the “**Choose Action**” menu, and select “**Apply Label**.” This will add the particular patch to your manual patch label. Keep in mind; unlike a patch smart label that can be set to get all present and future patches that meet specific criteria, the manual label will only receive the specific patches you add to it. It will not receive new patches over time. Because of this, a manual patch label is not advisable for general scheduled patching.

Patch Subscription Settings

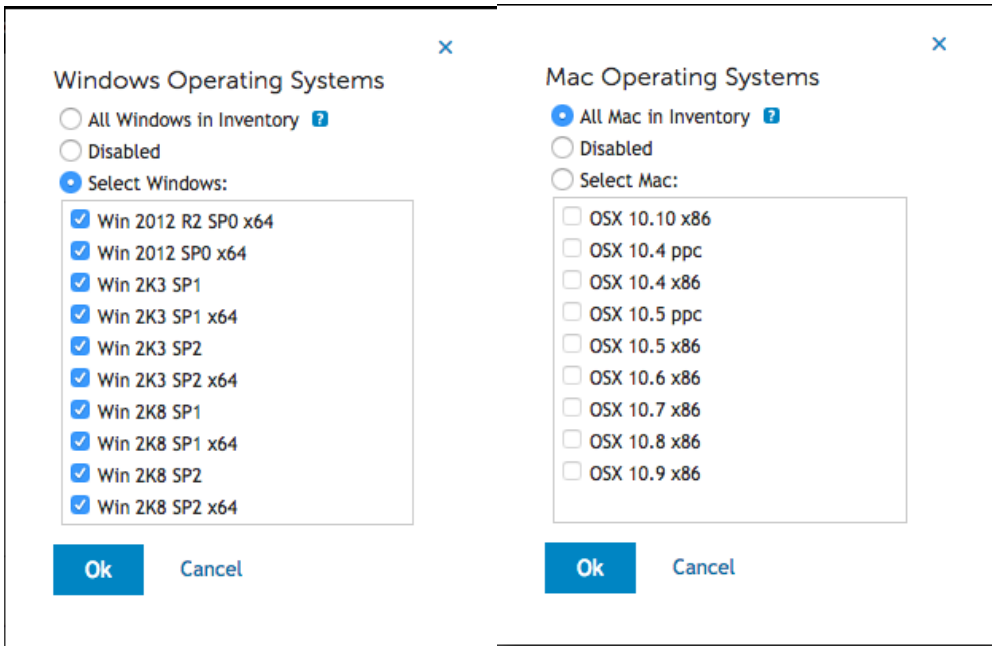
The patch subscription settings are controlled by selecting **Security -> Patch Management -> Subscription**. In this area, each Organization in the K1000 will select the operating systems, which they want to patch.



An example of a basic setup for patching subscriptions

The Org admins can select individual operating systems, “**All Windows in Inventory**”, or”“**All**

Macs in Inventory. Additionally, Org admins will select which languages and locales of the operating systems they want to patch. Most Orgs will select '**English**'.



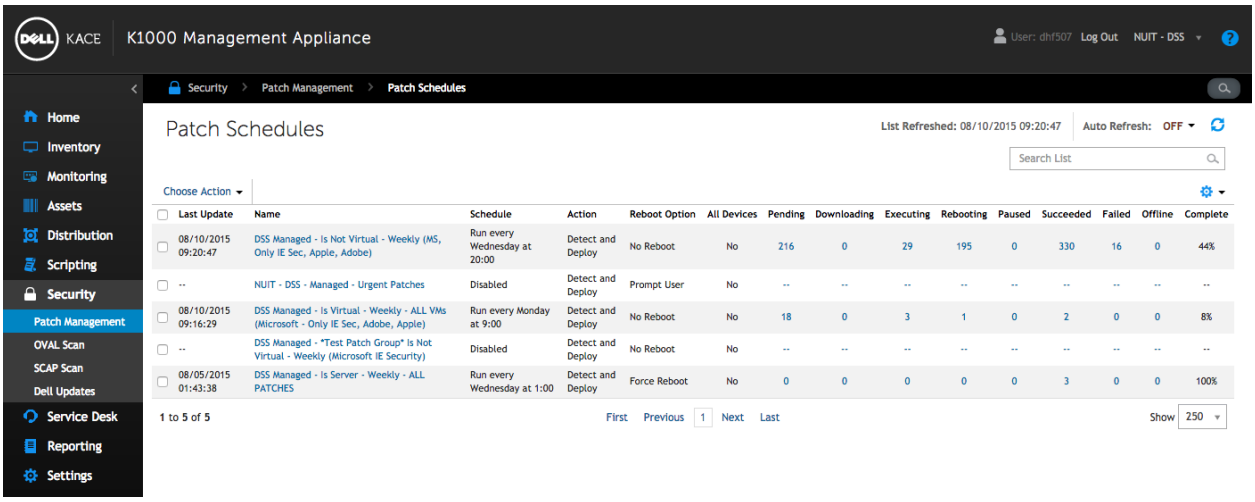
Detail: Operating system selection for Patch Subscriptions

Under the heading "Operating System Patches" it is best practice to select "**All Types**" and "**All Impacts**".

Under the "Application Patches" heading it is important to exclude "**Software Installer**" from the types. This will prevent unintended applications from being deployed to your computers, for example Microsoft Skype. For Publisher and Impacts it is recommended to leave it set at "**All Publishers**" and "**All Impacts**".

Patch Schedules - What, When, and How

Patch schedules allow you to define when to initiate Patch **Scans** and **Deploys**. It involves making trade-offs between inconvenience to users and the desire to have machines patched. Accessed from **Security -> Patch Management -> Schedules**. There is no one size fits all approach.



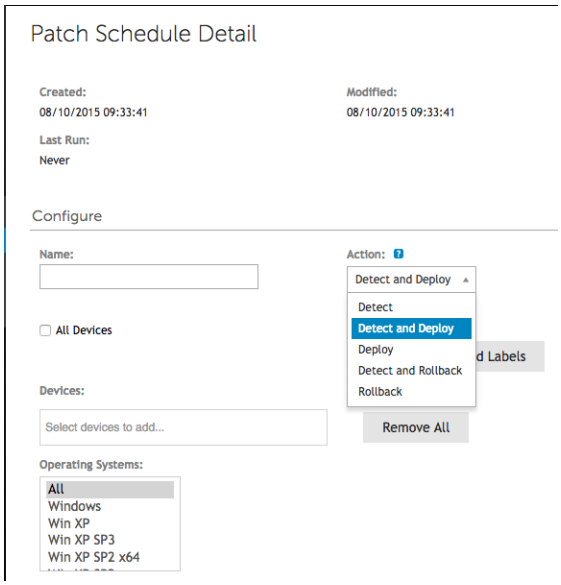
A sample showing multiple patch schedules in a single organization and the associated visible columns of details

Patch Actions

Creating a new Patch Schedule is accomplished through the “Choose Action” menu in **Security -> Patch Management -> Schedules** and selecting “New.”

When choosing clients to target, one must choose the action of the schedule. You have five action options: **1) Detect, 2) Deploy, 3) Detect and Deploy, 4) Detect and Rollback, or 5) Rollback .** First, “**Detect**” scans the computer for applicable patches, but will not install them. One can limit the sets of patches to scan for by choosing what patches to detect later on in the configuration. Next, “**Deploy**” installs patches that have been detected as needed, and is again limited to patch labels. And as one could expect, “**Detect and Deploy**” does both. This action first scans for patches, then deploys them. This process will repeat until all patches are detected as installed, or their installation has been reported as failed. This process will even continue after a machine reboot.

“**Detect and Rollback**” and “**Rollback**” are for rolling back patches, and work similarly to the methods described earlier. These are not options that we would use in setting up a preliminary patch schedule, but are items to be used when certain deployed patches need removal.

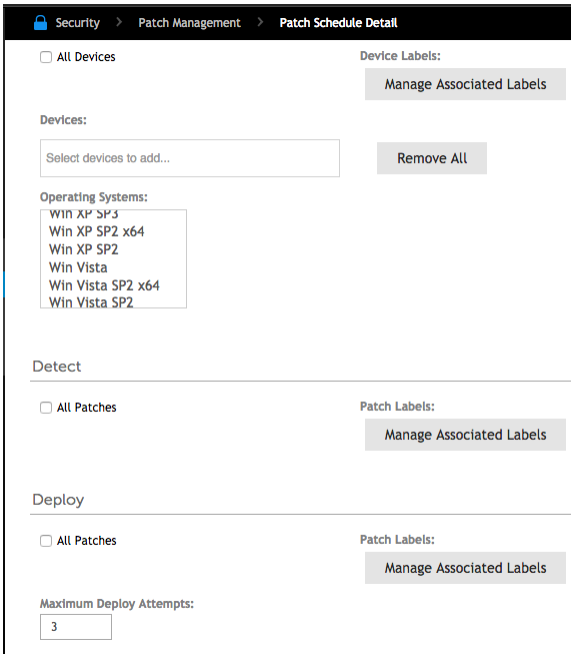


Detail: Setting the Patch Schedule action

Machine and Patch Selection

The next step of creating a Patch Schedule is to select what patches one wants to detect and/or deploy and to what machines to target. These are best grouped by labels (see **Machine Labels** and **Patch Labels** above for more information). Again, one should make logical groupings, such as: OS type, only Java patches, all Microsoft patches, or patches that have been released for more than 1 week.

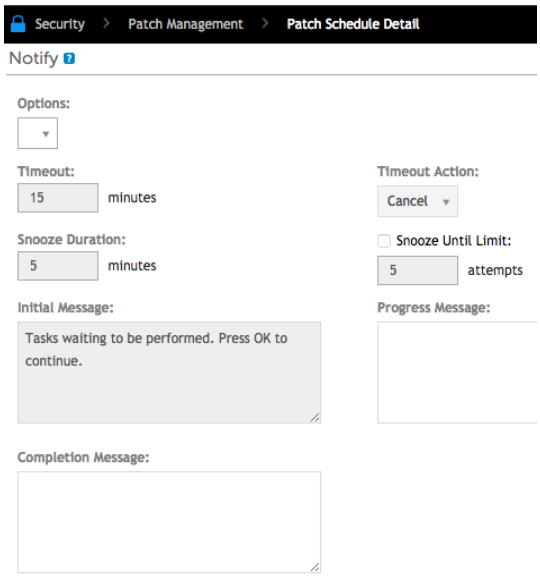
Machines are targeted (individually and/or via label) at the botton of the“**Configure**” section, while patch label selection is found in the following “**Detect”** and “**Deploy”** sections.



Detail: Configuring target devices and patches used in a Patch Schedule

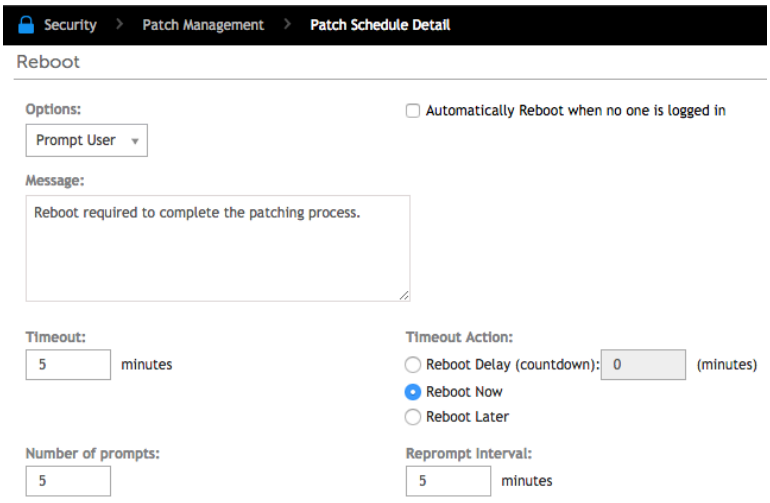
Notifications and Scheduling

"**Notify**" is the next set of options when setting up a Patch Schedule. With these options, you can choose to notify a user prior to any patching starts and allow them to defer the whole process. This is useful if one wants to allow users to defer the patching process to a more convenient time. Installing patches while the user is using the machine could possibly lead to system instability.



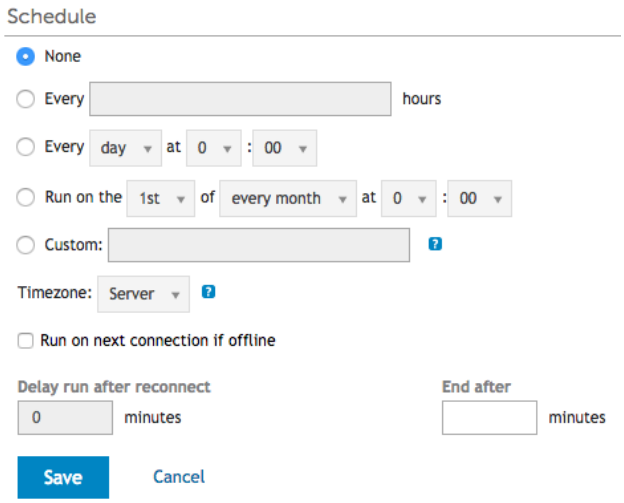
Detail: Set end-user notification options in the "Notify" section

The "**Reboot**" section is similar to notify. Users can be prompted to reboot a machine, and are continually re-prompted at a set interval until they comply. If no action is taken, the reboot can be forced or deferred.



Detail: Set reboot options in the “Reboot” section

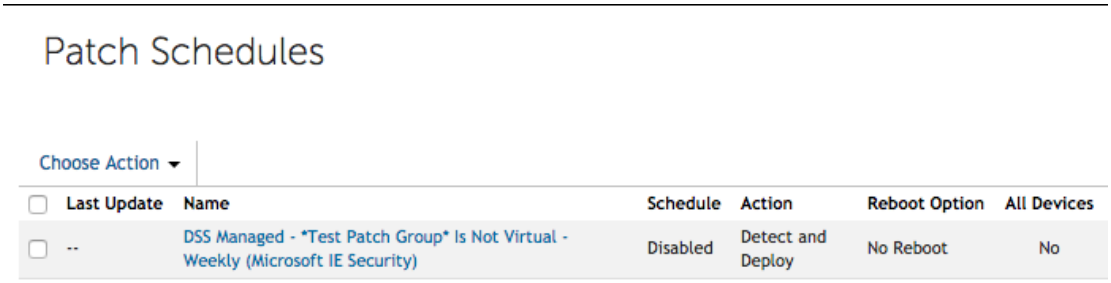
Finally, there is the actual schedule. Determine what schedule to apply updates, such as 3:00 am every Wednesday, or every 6 hours. There is one other important option – “**Run on next connection if offline**”. This means that if the computer was unavailable during the scheduled time, it will starting patching the next time it establishes a connection with Kace. Optionally, one can add a delay to allow a freshly booted machine time to start up before patches start being deployed. The default of 10 minutes has proved to be an effective and functional delay.



Detail: Set scheduling options in the “Schedule” section

Best Practices for Patch Schedules

Each schedule created should have a clear goal such as, “patch Java as soon as the patch becomes available in Kace,” or “patch the test control group computers with the latest patches every morning,” or “patch all client computers with patches that have been approved after testing in the test bed every Tuesday evening.”



An example test patch schedule

A take away learned in managing patching, is that users do not appreciate surprise reboots. Force rebooting a faculty member’s laptop during a lecture will cause a conflict. Forcing a reboot of all desktops once a week in the middle of the night is a more acceptable approach if it has been communicated to users to expect it.

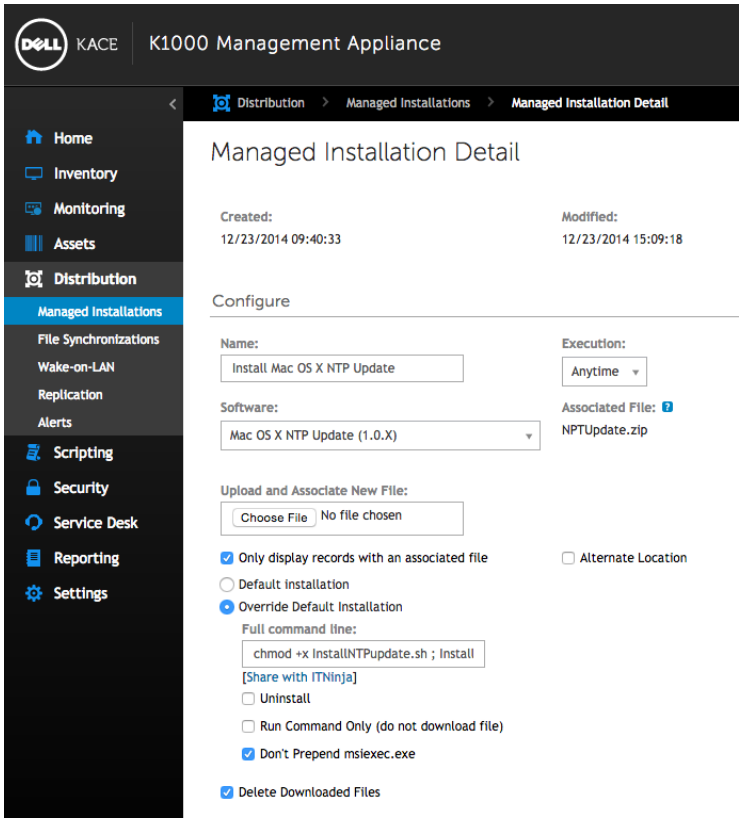
Another issue that has occurred is a case where a particular patch keeps trying to install, but

fails deployment. After each attempt, the user is prompted to reboot. This can lead to users complaining that Kace kept prompting for reboot. A workaround is to set patches to only try installing once.

Ad Hoc Patch and Zero Day Attacks:

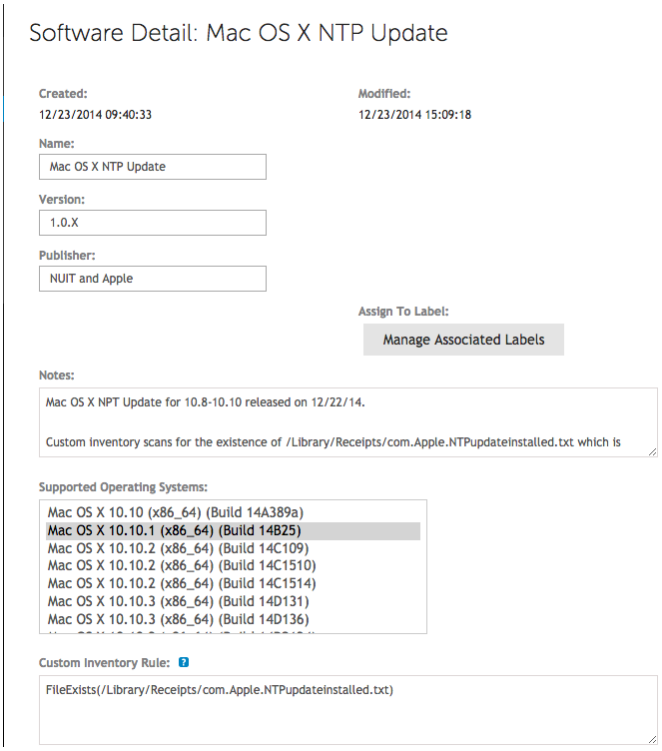
The standard patch feed in Dell Kace generally populates several days behind when a vendor releases a patch. So what happens when you need the patch to be installed before it becomes available in Kace? The answer is to write a managed distribution or a script for manual deployment of a patch. You need to weigh the risk of going un-patched versus the extra disruption to users and the time to build and test the patches. In most cases, waiting will be the better choice. Kace usually has patches available for Windows, critical Java, and Flash within a couple days of manufacturer release. If the computers that need patching hold sensitive information, creating your own patches and pushing them through managed installations or scripting may be advisable.

For software that is not patched through Kace patching, managed installations, scripting, or relying on the manufacturer's auto-update system are the only options. When creating a manual patch, consider the same issues of targeting and timing as regular patches.



Detail: Sample Mac OS X NTP patch Managed Installation (created because patch was not yet available from Apple)

Above is an example of a manually made patch for the OS X NTP vulnerability that was created to patch OS X prior to the NTP update being available through Apple's built-in OS X patching. A Managed Installation is a great way to deploy one-off software patches, but keep in mind that you may need a way to track that a manually-deployed patch has been installed with a **Custom Inventory Rule**. The creation of this rule will depend on how the patch installation was built.



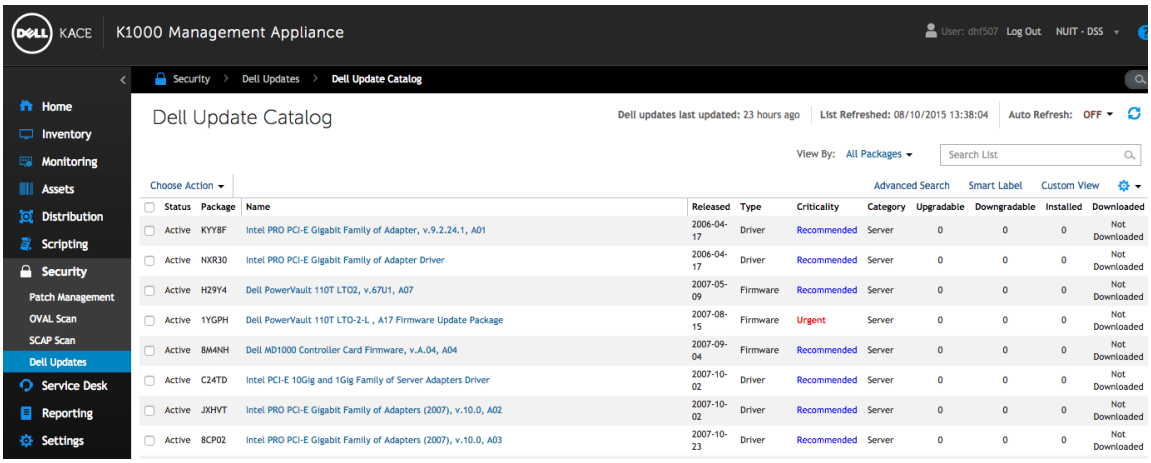
Detail: Mac OS X NTP patch installation detail with Custom Inventory Rule

Using a **Kace Script** to deploy an ad-hoc patch will not require the creation of a custom inventory rule, but has a caveat of limited tracking and deployment scheduling options.

Informing users about manual patch is advisable. Both scripting and managed installation allow for messages before, during, and after running. However, this method lacks the option of prompting for reboots. If a reboot is required, users must be notified prior to deployment.

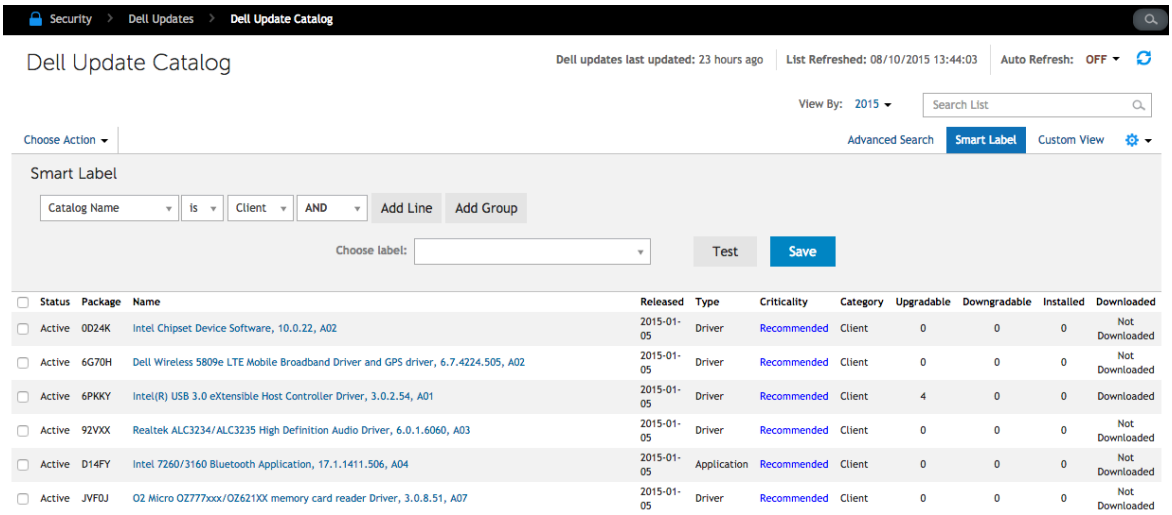
Dell Updates:

Similar to software patching, Kace allows a managed approach for applying device updates to Dell machines. New Dell Updates are downloaded on the K1000 daily. Accessed from **Security -> Dell Updates**, this area allows you to keep Dell hardware updated with the latest manufacturer drivers, firmware, Bios, and device software. To view the types of Dell Updates available, select **Security -> Dell Updates -> Catalog.**



The Dell Update Catalog

From **Home -> Label Management -> Smart Labels**, one can build a **Smart Label** to filter on specific updates you wish to deploy from the “Choose Action” menu and selecting **New -> Dell Package Smart Label**. Note, software updates may include install or upgrades to Intel hardware utilities (wifi, trackpad, etc.) found on most Dell machines. Dell Updates most likely will require machine reboots and sometimes display verbose messaging to end users.

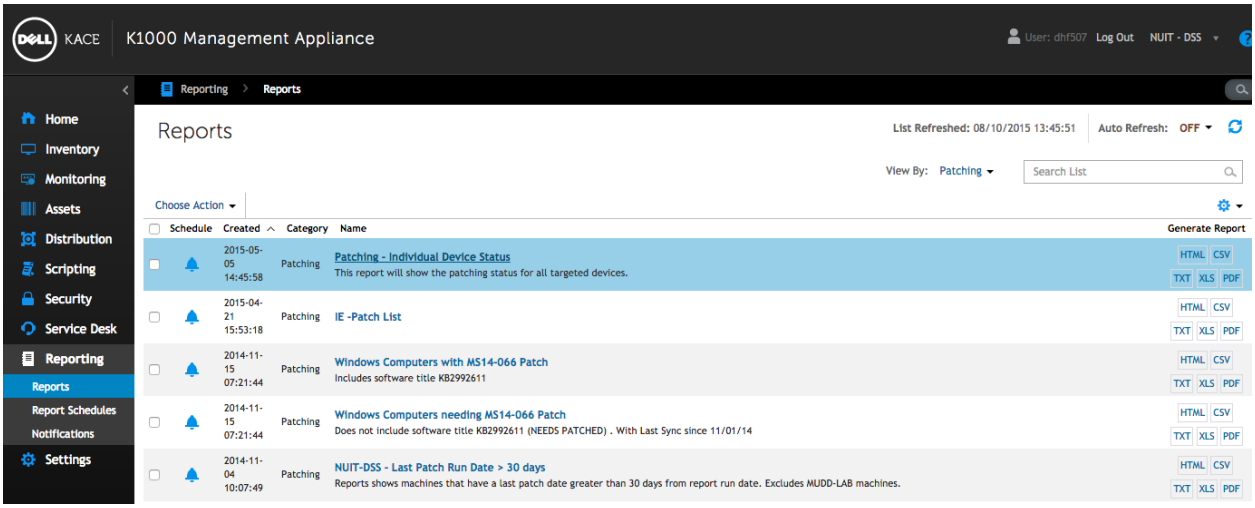


Detail: Creating a Dell Update Smart Label

Dell Updates are controlled and scheduled identical to Patch Management from **Security -> Dell Updates -> Schedules** allowing you to limit the types of updates (smart label), target specific machine groups (machine label) and when the update should occur (notification and scheduling). Important: if your support area would rather handle Dell device updates on an ad hoc basis, it is recommended to at least configure a “**Detect**” schedule within Dell Updates so that you can utilize Kace reporting to maintain a current record of machines as they relate to Dell device updates.

Reporting:

Dell Kace reporting allows Org admins to create and run reports to assist with determining patch status on machines, machines requiring reboots to finish patching, and other inventory related details. Reporting can also be configured to be delivered to specific email addresses each morning or ran ad hoc to further determine which machines require further review relates to patch management. Access patch reporting from **Security -> Patch Management -> Reporting**. When creating a new Kace report, specifying it as a "**Patch**" report, will allow it to be included in this navigation shortcut.



The Reports section highlighting a patching report

Reporting is also available for Dell Updates. Found under **Security -> Dell Updates -> Reporting**, one can do comparisons on computer inventory vs. available Dell updates. If needing a new report for either patch or Dell updates, using the **Kace Reporting Wizard** or manually creating the report using preconfigured SQL syntax allows groups to further customize patch reports to their liking.

Appendices or Related Information:

- Essentials of Patch Management Policy and Practice
- Patch Management Mailing List | Subscribe to list
- SANS Institute - Patch Management - part of standard operations...
- Best Practices for Applying Service Packs, Hotfixes and Security Patches
- IT Ninja website: Software Library | Questions & Answers | Blog
- Microsoft Security Response Center Blog
- Microsoft Security Research and Defense Blog

- Microsoft Office Updates Blog
- Adobe Product Security Incident Response Team (PSIRT) Blog
- Oracle Software Security Assurance Blog
- Apple Product Security
- Apple Security-announce mailing list
- SEC_UNITS Mailing List

Original Issue Date:

04/17/15

Revision Dates:

8/10/15 - Revised by Daniel Friedman. Added screenshots/captions, supporting edits

---

[1] For simplicity, both commercial software and open source software are considered to be provided by vendor.