Advanced Patch Lifecycle Management with SUSE Manager

Methods and approaches for managing updates in multi-landscape environments

SUSE Manager

Jeff Price, Principal Architect (SUSE)

Advanced Patch Lifecycle Management with SUSE Manager

Methods and approaches for managing updates in multi-landscape environments

This document describes how to set up and configure a SUSE Manager implementation to enable companies in the delivery of often requested features. Covered are the automatic creation and archive of patch sets by a defined time period, a consistent method of patch promotion and delivery through numerous landscapes and environments, and the exception process for handling patches that need to be excluded from a patch cycle. Also explained are the creation of an environment with historical patch sets, the minimal need for host channel subscription manipulation from cradle to grave, and the support for service pack migration with custom child channels.

Disclaimer: Documents published as part of the SUSE Best Practices series have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

Contents

1 Overview

For many organizations today, keeping systems patched and secure is one of the biggest ongoing challenges that systems administrators can face. Both proprietary and open source software companies are constantly at work providing updates that fix flaws discovered in their software products.

Adding to the pressures are the requirements imposed by compliance initiatives like PCI-DSS (Payment Card Industry Data Security Standard), SOX (Sarbanes-Oxley), HIPAA (Health Insurance Portability and Accountability Act), FIPS 140 (Federal Information Processing Standard), etc. Included in most of these "standards" are rules that dictate the frequency or time frames for patching and often these schedules are modified because of the severity level of a specific patch.

Businesses large and small are depending on an ever-changing landscape of applications on a wide range of computer infrastructure. These can include physical and virtual hosts and may have complicated application stacks. These applications may have specific kernel version dependencies, have strict uptime requirements or be tied to a specific run time environment (like Java). They may only be certified on a specific Linux OS release, or have a particular set of configuration requirements. There might be extra hardening or platform requirements that set these hosts apart from the rest.

Additionally, many companies have created duplicate host environments like "development" or "QA" landscapes that mirror "production" to allow for testing or validation of changes. This means at least one more complete set of hosts to manage, patch, and report on. This increases the challenges of meeting a rigorous patch schedule that often requires patching all hosts on a quarterly basis (or more frequently for critical vulnerabilities).

Is there any way to address all of the issues and regain control of a complicated enterprise Linux environment?

1.1 A SUSE Solution

SUSE Manager delivers best-in-class Linux server management features like automated software management, system provisioning, and monitoring. It provides a unified Linux management infrastructure by standardizing and automating server patching and management. These tasks can be performed faster and with fewer errors, which improves IT staff productivity while reducing server downtime.

1.1.1   Definitions

This document uses the word **landscape** to describe one of several groups of Linux hosts. This is commonly used with enterprise software deployments as a way to define the role/quality of a host. Examples include production (PROD), development (DEV), quality assurance (QA), user acceptance/testing (UAT), etc.

The word **environment** is also used. In this document, environment is defined as a physical separation of Linux hosts, like Corporate (CORP), Store (STORE), or NPE (Non-Production Environment).

Lastly, the word **organization** is used. SUSE Manager has a concept of organizations and defines them as a separate management environment placed under the default organization that is created during initial installation of a SUSE Manager Server. This is a single parent (like an LDAP "O") with many child placements (like an LDAP "OU"–restricted to a flat OU structure with no nesting). Organizations in SUSE Manager will typically have separate credentials required to administer them via the Web UI–and any Linux hosts managed by SUSE Manager will be registered into a single organization and only visible there. Keep these definitions in mind as you read.

1.2   Example Scenario

The Chameleon Corporation is the world's largest reptile and amphibian pet store, and has its headquarters in Yemen. It has more than 1500 stores operating under the names LizardsRUs, UnVeiledInc and Chameleo-rama in five different countries.

The Chameleon Corporation has the following requirements for managing patch and security updates to their Linux hosts. Because of compliance constraints, their patch lifecycle (the timing and delivery) is based on their corporate security policies, the host environments/landscapes, and their future management needs.

Here is a high-level summary of the requirements:

1. The published Security Policy states the need to patch all Linux systems with the latest available updates, bug fixes, and enhancements on a quarterly basis. The quarters are defined as Q1: January 1st through March 31st, Q2: April 1st through June 30th, Q3: July 1st through September 30th, and Q4: October 1st through December 31st.

2. The published Security Policy also states the need to patch all Linux systems with "critical" security updates (CVSS score of 7 or greater) within one month of release of the patch. This introduces a second schedule that may overlap any existing patch roll out.

3. The company has three different host environments: Corporate, Store, and NPE. These host environments are partitioned using SUSE Manager organizations. Each SUSE Manager organization has visibility into the default SUSE Manager organization's software channels because of trusts, yet the managed hosts are naturally grouped and managed by their environment.

4. There are at least three landscapes in each environment: DEV, UAT, and PROD. There are additional landscapes for some environments (for example, SIT, DIT, etc.) but these are handled similarly to the aforementioned landscapes. Patch lifecycle promotion simply accounts for moving patches into additional stages as needed.

5. A "patch exceptions" process exists, as there is often a need to account for the removal of patches (for example kernel updates) from a rollout. Some hosts might have hardware drivers or third party software that depends on specific kernel versions. This exception process would be handled on a case-by-case basis.

In summary, these requirements highlight the need for two patch schedules, across three (or more) landscapes, in three SUSE Manager organizations, with a way to handle a storage location for patches that cannot be rolled out (exceptions), yet need to be tracked, re-mediated and reintroduced to ensure compliance.

SUSE Manager can handle these complexities very well and can be set up to allow flexible delivery of these requirements. Automation can even be applied to assist with some facets, and procedures can be developed to ensure the consistent delivery of patches that meet the published company policies.

1.3 Features

This document describes how to set up and configure a SUSE Manager implementation to enable companies to deliver these often requested features:

- Automatic creation and archive of patch sets by quarter (or any other time period)

- A consistent method of patch promotion and delivery through numerous landscapes and environments

- An exception process for handling patches that need to be excluded from a patch cycle

- Environment creation with historical patch sets

- Minimal need for host channel subscription manipulation from cradle to grave

- Support for service pack migration with custom child channels

Many of these features can be tuned to meet a change in requirements, like an increase in, or a reduction of, the number of landscapes (DEV and PROD or Sandbox, DEV, QA, UAT, PRE-PROD, PROD, etc.), or a more, or less frequent delivery calendar. Keep this in mind as you read. SUSE Manager is a very flexible product that has helped many customers create and deploy solid systems and patch management frameworks. Your requirements may differ from the ones described here, but the flexibility of SUSE Manager enables you to meet your ongoing and changing needs.

1.4 Guidance

This guide is based on SUSE Manager 2.1, but can also be used with SUSE Manager 3.0. The document will not highlight any Salt minion specifics nor should this matter change how channel organizations and patch manipulation are handled here.

A fresh installation is not necessary, but some assumption is made that you have a working SUSE Manager environment, with some registered hosts, and a decent knowledge of the terms and capabilities. The SUSE Manager documentation at https://documentation.suse.com/suma/3.2/ ↗ is an excellent source of information that is often updated. Refer to the documentation for clarification of any process that seems confusing. Many of the implementation steps detailed in this document are enhanced examples based on procedures, capabilities, or step-by-step "how-tos" in our product documentation.

Note: Test Your Implementation

Some advanced features (like using the `spacecmd` or the XMLRPC API) will be explained in more detail, but these explanations may not be exhaustive. We will include some code examples in the Appendix that can be used as-is, or modified to suit your particular environments or needs. However, these come with **no warranty**, and should not be expected to be perfect for any implementation. **Always test** your particular implementation before any production usage.

The next section will describe the setup of SUSE Manager to meet the requirements defined in the **Scenario** section. We will start with a description of the architecture (and components) developed to meet the requirements, and then proceed with steps to create it yourself. Each

component within this guide is used together with the others to deliver a comprehensive solution in the specific scenario described herein, but might be used individually to help you meet your own solution requirements.

2 Implementation

Let's review. We need to set up SUSE Manager and create a set of procedures, processes, and/or tools (scripts) to help us with our patch lifecycle management. This will include the creation (and optimally an "automatic" creation) of patch archive channels from the SUSE provided "updates" channels for the versions of SUSE Linux Enterprise Server we use. These patch archives will be the sources for our patch promotions, but can also be a source to set up testing environments based on a patch set rolled out in the past. This could be used to troubleshoot an error condition for example, by creating a lab full of hosts patched up to any previous patch set. We can also demonstrate and visualize a clear view of compliance defined by a time-stamped set of patches.

We also need to create a set of software channels that will get our updates when **we** decide to roll them out. Each landscape needs to receive them from the previous "validated" landscape **only**. This means they will first get put into DEV, and are tested on the machines there. Then they move into QA and get tested there. Finally they get moved into PROD. The whole thing will start over every quarter, and we will handle exceptions if and when they occur. Every exception will be tracked by another established process with the constant goal of re-mediating any exception and making sure the patch is reintroduced as quickly as possible to ensure our clear view of compliance.

Other interruptions to a quarterly roll out might be the release of a "critical" security patch that needs to be deployed more rapidly than once-a-quarter. To handle this extra schedule, we should create an addition channel for any critical patches to track them and their roll out. Patches that should be considered as critical patches are any CVE that has a CVSS score of 7.1 or greater, or otherwise that is mandated by our corporate security organization. We have two different SUSE Linux Enterprise Server versions that we currently manage: SUSE Linux Enterprise Server 11 SP3 and SUSE Linux Enterprise Server 12. We want to plan for using SUSE Manager to assist with service pack migrations, so setting up to account for SUSE Linux Enterprise Server 11 SP4 initially would be ideal. We also maintain both 32-bit and 64-bit Intel/AMD architecture versions of SUSE Linux Enterprise Server 11 SP3 so we will need to account for both of them.

There are two different environments we need to manage: Corporate (CORP) and Store (STORE). We have created separate SUSE Manager organizations to allow our systems administrators an exclusive view of the hosts they are responsible for. Each SUSE Manager organization has the three landscapes (DEV, QA, and PROD) that they use for patch testing/promotion.

Activation keys and bootstraps should be created for each SUSE Manager organization (each environment) and each landscape so that hosts can be registered into their appropriate, and most likely, permanent role. For example, a bootstrap script will register a SUSE Linux Enterprise Server 11 SP3 (64-bit) DEV host into the correct organization and will add the appropriate base and child channels that should never need to be manipulated again. Automation is preferred. As mentioned, the use of scripts or scheduled jobs would greatly enhance the solution and provide relief to the administration staff.

2.1 Architecture

Here is a high-level outline of the architecture and channels:

Note: Use of Prefixes

For any **Base** channel listed here–this can be a clone tree of the SUSE vendor channels–as such it would include a *prefix-* in the name. This can aid the handling of service pack migrations described in the channel creation sections that follow.

- SUSE Manager Organization

    - Default (Main) - 1

        - 32-bit Archive Channels (Public)

            - Q2 - 06-30-2015 - SLES 11 SP3 Updates for i586

            - Q3 - 09-30-2015 - SLES 11 SP3 Updates for i586

        - 64-bit Archive Channels (Public)

            - Q2 - 06-30-2015 - SLES 11 SP3 Updates for x86_64

            - Q3 - 09-30-2015 - SLES 11 SP3 Updates for x86_64

- Q2 - 06-30-2015 - SLES 12 Updates for x86_64

- Q3 - 09-30-2015 - SLES 12 Updates for x86_64

- Clone Sets or Child Channels within SUSE Base Channels

- Base : SLES 11 SP3 Pool for i586

  - DEV - Current Patch Set - SLES 11 SP3 Updates for i586

  - QA - Current Patch Set - SLES 11 SP3 Updates for i586

  - PROD - Current Patch Set - SLES 11 SP3 Updates for i586

  - Patch Exceptions (DO NOT SUBSCRIBE) - SLES 11 SP3 i586

  - Security ASAP Exceptions - SLES 11 SP3 i586

- Base : SLES 11 SP3 Pool for x86_64

  - DEV - Current Patch Set - SLES 11 SP3 Updates for x86_64

  - QA - Current Patch Set - SLES 11 SP3 Updates for x86_64

  - PROD - Current Patch Set - SLES 11 SP3 Updates for x86_64

  - Patch Exceptions (DO NOT SUBSCRIBE) - SLES 11 SP3 x86_64

  - Security ASAP Exceptions - SLES 11 SP3 x86_64

- Base : SLES 12 SP3 Pool for x86_64

  - DEV - Current Patch Set - SLES 12 Updates for x86_64

  - QA - Current Patch Set - SLES 12 Updates for x86_64

  - PROD - Current Patch Set - SLES 12 Updates for x86_64

  - Patch Exceptions (DO NOT SUBSCRIBE) - SLES 12 x86_64

  - Security ASAP Exceptions - SLES 12 x86_64

- Patch Exceptions Channels (as above)

- Per Base Channel - 1 channel for each "Updates" channel:

  - Examples:

    - Patch Exceptions (DO NOT SUBSCRIBE) - SLES 11 SP3 i586

    - Security ASAP Exceptions - SLES 11 SP3 i586

- Corporate (CORP) - 2

  - Any public channel will be visible here

  - Private channels can be shared per organization

- Store (STORE) - 3

  - Any public channel will be visible here

  - Private channels can be shared per organization

2.2   Automation and Process Design

Automation will include automatic channel clone creation for the Archive channels. To best suit a particular company's use cases and product usage, a combination of scripts (cron and Bash) and a configuration file will be used. The archive creation will be triggered by a custom cron job as follows. Criteria for dates will determine the exact setup. For example, a quarterly schedule could be implemented based on the first or last day of a given quarter. If the trigger time is the **end** of a quarter, it would need to consider day 30 on the 3rd and 9th months of a calendar (March and September), and day 31 on the 6th and 12th months (June and December). This will require two crontab entries such as (cron syntax):

```
0 0 30 6,9 * /path/to/archive-create-script
```

```
00 0 31 3,12 * /path/to/archive-create-script
```

If the trigger for the quarter is going to happen on the 1st of a month, then only one crontab entry would be required, for example:

```
0 0 1 1,4,7,10 * /path/to/archive-create-script
```

Advanced Patch Lifecycle Management with SUSE Manager

The actual "archive-create-script" will also need logic in it to determine the quarter it was run and formatting logic to provide the correct syntax for SUSE Manager channel cloning, etc. The script should also loop through each architecture and SUSE Linux Enterprise Server version to make a full set of archives based on the company's choices. Processes for patch promotion will be created so that administrators can precisely and explicitly define when a patch set is moved **into** a given landscape, and just as importantly, from **where**. The easiest method to do this is using a script that leverages the XMLRPC API exposed in SUSE Manager. The API contains "softwareChannel" methods called "mergePackages" and "mergeErrata". These two methods can be called with a source channel and target channel to quickly take content from one channel and merge the differences into another.

The last process will define a workflow for removing a patch from the promotion process–and making sure it is stored to be tracked. These patch exceptions can later be reintroduced to a patch roll out calendar as the reason for the exception is re-mediated. In addition to the exception process of **removing** a patch, there is a similar process for **adding** a patch as an emergency dictates.

Emergency patches will likely always be available in the current SUSE Updates channels, but might not have been introduced into an archive channel yet. As the quarterly archive channel creation job is triggered, any available patches would be part of an archive set, however at any other given point in time, this might have occurred before the patch was released or it may be up to three months until a new archive is created that contains this recent patch. A method for searching and then copying a patch into a target channel will need to be created. This will handle those times when a critical patch is released by SUSE but your archive channel will not get a copy until the quarter is up.

Copying these emergency patches into place does not affect the normal deployments or creation of archive channels. The next archive channel creation will also contain this patch, and the use of an emergency patch channel with a duplicate patch (in the case of merging the same patch later) will not cause any conflicts with a system that already has installed it.

Further explanation can be found below in *Section 2.4, "Exception and Escalation Channels"*.

2.3 Patch Promotion Cycle

Understanding how patches (and the packages they include) are obtained from SUSE, where they go after download, and how they are arranged within SUSE Manager is key to setting up the Advanced Patch Lifecycle system. By default, SUSE Manager arranges its channel sets by a particular version and architecture of SUSE Linux Enterprise Server and its extensions.

For example, a base set of channels within SUSE Manager for SUSE Linux Enterprise Server 11 SP3 x86_64 includes the parent or base channel called a "pool" channel. This channel contains packages that are the functional equivalent of the installation DVD for SUSE Linux Enterprise Server 11 SP3 64-bit. Under this pool channel are all of the "child" channels which include a channel for the SUSE Manager Client packages, called SUSE Manager Tools channel, and an updates channel called, in this case, SUSE Linux Enterprise Server 11 SP3 Updates for x86_64.

A normal set of channels that a managed host subscribes to will include at least a parent channel (pool), a SUSE Manager Tools channel, and an Updates channel. In some cases like SUSE Linux Enterprise Server 11 SP2, this will include the SUSE Linux Enterprise Server 11 SP1 Pool, the SP1 Updates channel, the SP2 Core channel, SP2 Updates, SP2 Extension Store, and the SP2 SUSE Manager Tools channel.

Note: Long Term Service Pack Support (LTSS)

Because both SUSE Linux Enterprise Server SP1 and SP2 are in their LTSS phase, there are no longer any non-LTSS updates being created from SUSE. This means that any SP1 or SP2 hosts a company might have will need to purchase LTSS and use the LTSS Updates channel to receive further patches and support. Visit http://www.suse.com/lifecycle ↗ for more information.

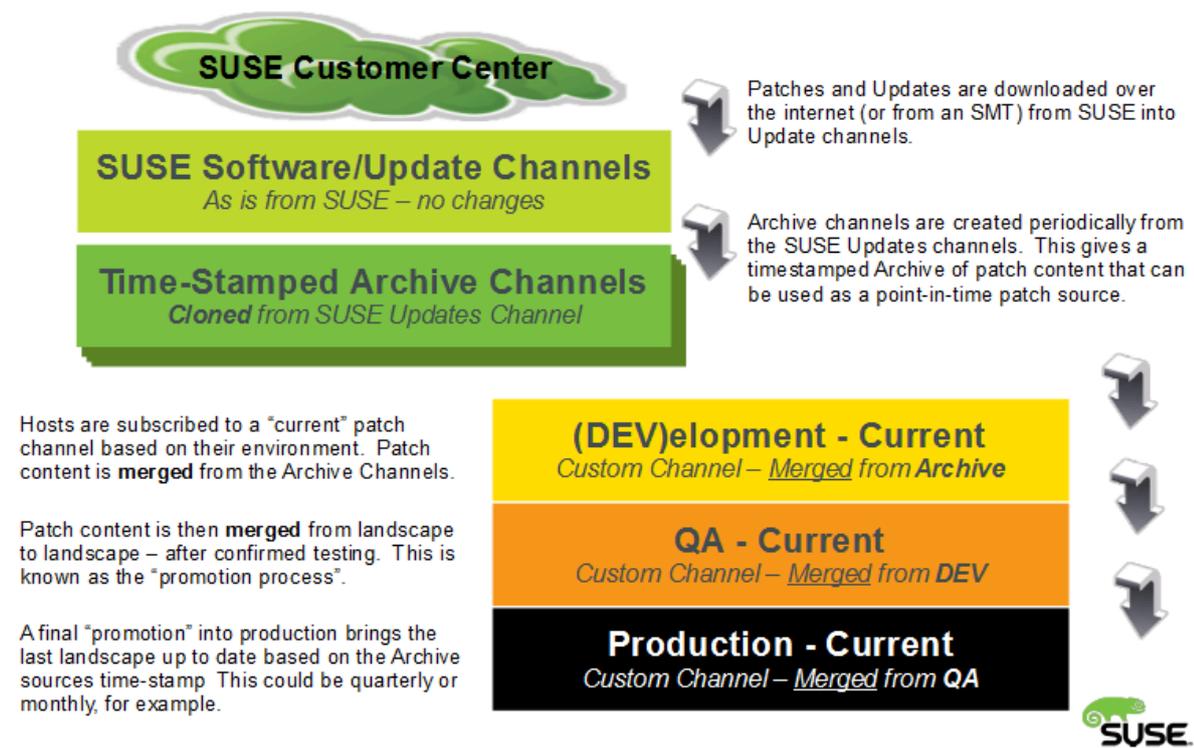Here is a graphic that illustrates the patch promotion process:

FIGURE 1: PATCH PROMOTION PROCESS

2.4 Exception and Escalation Channels

There might be several reasons to delay or remove a patch from a deployment cycle. A patch could be causing issues with the stability of a system or systems, it might not have passed a testing cycle from the current or previous landscape, or there could be some other reason not to deploy it within a patch cycle. On the other hand, there might be a reason to deploy a critical patch (one that patches a security vulnerability or a debilitating bug) ahead of schedule. Because of the critical nature of the patch it needs to go through testing and deployment as fast as possible– without waiting for the next quarterly patch cycle.

To handle the periodic removal or addition of these exception patches, processes should be created to track them appropriately. While these exceptions are being tracked, they will be copied into their own channel containers for safekeeping, accounting, and reporting.

These channels can be created within the Archive base channel or within the cloned base channel for the version of SUSE Linux Enterprise Server that corresponds to the patch exception (recommended).

The process for excluding a patch will be to first **copy** the patch (and associated packages) into the Exception Channel, and then to **remove** the patch (and associated packages) from the current deployment landscape. Optionally it can be removed from the current quarter's archive channel but it is important to realize that the patch will be included in the following quarter's archive as well. It might be tempting to remove it from the SUSE Updates channel, but this can often lead to a mishandling or misreporting of compliance. It is much better to track and **resolve** the reasons for the exception in the first place. This way, the patch can be reintroduced to the patch deployment workflow and re-mediate any potential security issues or code flaws that the patch was created to fix.

2.4.1   Security Escalation Channel and Process Definition

The process for **adding** a security patch is similar to the process of excluding one. The key difference is that the patch needs to be copied into the Security Exception Channel from either the SUSE Updates channel or the latest archive if it has been created but not used for a promotion cycle yet. Assuming that this **new** critical security patch has been released in the middle of a roll out phase (you have already been promoting the current quarter's patches and this new one comes out), it is likely that this new patch has not yet appeared in an archive.

Another assumption is the need to test this new patch by promoting it through the landscapes as any normal patch set, but if necessary it can be copied directly to any stage. Simply **copy** this security patch into the initial landscape (for example DEV - Current) and follow the normal deploy-test-promote cycles to get it into production. Depending on the critical nature of the patch, the normal schedule for testing could be accelerated or in rare cases bypassed altogether assuming acceptable risk.

2.5   Components

Here is a list of things we will create to enable Advanced Patch Lifecycle Management using SUSE Manager 2.1:

Channels

1. Archive Channels

2. Current Updates Channels

3. Exception Channels (Patch Exclude or Security-ASAP)

   4. Optional SP-Migration Clone Set

Activation Keys / Bootstraps

   1. Per-Organization/Per-SLES version Activation Keys

   2. Per-Org/Per-SLES version Registration Bootstrap

Scripts

   1. Archive Creation Script

   2. Archive Sources List Control File

   3. Merge Tool Python Script

Crontab Entries / Automation

   1. Quarterly (or other frequency) archive creation call

2.6   Channels

As described in the architecture section above, several custom channels will be created to support the patch promotion process. These channels will store the archives (created as clones of the SUSE Updates channels) and they will also store the exception patches. For each landscape a custom channel will be created to hold patches that have been promoted. These channels will contain more and more content over time as patches are promoted each cycle.

For example, during Quarter 1 (Q1) the archive channel contains 159 patches, and these are merged into the DEV updates channel for SUSE Linux Enterprise Server 11 SP4. As SUSE releases patches in an ongoing fashion, when the next quarter's archive is created, Q2 for this example, it contains all of the Q1 patches and an additional 130 patches for a total of 289. The merge function will take the 130 new patches (and associated packages) and add them to the DEV updates channel during the next patch cycle.

Each new patch cycle updates the available patches in a given landscape's "current updates" channel. In this way each subsequent patch promotion adds the new updates and these can be applied to any hosts subscribed to the channel.

The exception channels will be populated by managing individual patches, copying them from one of the landscape updates channels or from one of the archive channels. When a patch is copied successfully into an exception channel, it can be removed from an updates channel or an archive.

2.6.1   Creating the Channels

Archive Channels:

Custom channels are created in SUSE Manager as "new", blank, and empty channels, or "clones". Technically, cloned channels can also be created as empty, but in this case we will create the archive channels with content sourced from the SUSE Updates channel, the patches, AND the packages they reference. Each archive channel will reside within an empty base channel. Base channels will be created for each processor architecture managed by SUSE Manager, for example, IA-32, x86_64, s390x, PPC, etc. Multiple archives matching the different versions of SUSE Linux Enterprise Server can exist in the base channel, but they will all be the same processor architecture.

Create the base archive channels:

For each processor architecture used by the hosts that SUSE Manager manages, create empty channels named "32-bit Archives Channel" and "64-bit Archives Channel":

1. In the Web UI click "Channels" then the submenu "Manage Software Channels" and click "+ Create Channel".

2. The channel label **must be** kept in the format of "architecture-patch-archives-channel", for example the 64-bit channel label should read "x86_64-patch-archives-channel".

   > ❗ Important: Label the Channels
   >
   > Failing to label these channels appropriately will cause the example archive script to not work properly (without modifications). SUSE Manager is very strict about channel "labels" consisting of lowercase letters, starting with a letter (not a number) and containing no spaces, etc.

In this example of creating a label starting with "x86_64" it is assumed a 32-bit channel might be "i586" or similar.

3. These archive channels have "NO PARENT", because they "ARE" parent channels.

4. Select the appropriate architecture for this channel: x86_64 for 64-bit, and IA-32 for 32-bit, etc..

5. Select the Public radio button for these channels to ensure their visibility to all organizations.

6. Repeat for each architecture type of SUSE Linux Enterprise Server you have deployed.

Create the cloned archive channels.

For each version of SUSE Linux Enterprise Server that is used, create cloned channels:

1. In the Web UI click "Channels" then the submenu "Manage Software Channels" and click "Clone Channel".

2. Select the SUSE "Updates" channel of the SUSE Linux Enterprise Server version you want to create an archive of as the Source.

3. Make sure to select the radio button for all content/patches.

4. Name the channel appropriately with an indication of Time/Date and what the archive contains, for example, "Q3 - 09-30-2015 - Archive of SLES 11 SP3 for x86_64".

5. Create a label using proper syntax, for example:

```
q3-09-30-2015-archive-sles11sp3-x86_64
```

6. Important: select the appropriate base archive channel to contain the clone.

7. Include an appropriate description of the channel in the Summary and Description fields.

8. Click "Create Channel" when finished.

9. Scroll down and select the Public radio button and then click "Update Channel".

10. Repeat for each SUSE Linux Enterprise Server version under each architecture.

Optional: Use spacewalk-clone-by-date to create archive channels from the past:

See the Appendix for an example source file that can be used with the spacewalk-clone-by-date utility. This configuration file can be used to create a clone channel with a certain date range as a filter for the content in the channel:

1. The utility is called with the following syntax:

```
spacewalk-clone-by-date -c "config file"
```

2. Repeat this for each archive channel with a past date, modifying the configuration files to match the appropriate dates and channel source/target names.

2.6.1.1 Service Pack Migration Support with Update and Exception Channels

There are a couple of generally accepted methods to manage a set of channels that allows consistent and expected behaviors when leveraging the service pack migration feature. You can find this feature in the software tab of a particular host. It allows you to update the service pack of a particular host in a one-click fashion. When using the patch lifecycle methods described in this document, some challenges arise when using the service pack migration feature. When you select a given host to do a service pack migration, the UI proposes (dictates) a set of mandatory "child" channels that will be subscribed to for a migration action. These mandatory child channels (grayed-out in the UI) will normally include the SUSE provided update channels for a given version of SUSE Linux Enterprise Server.

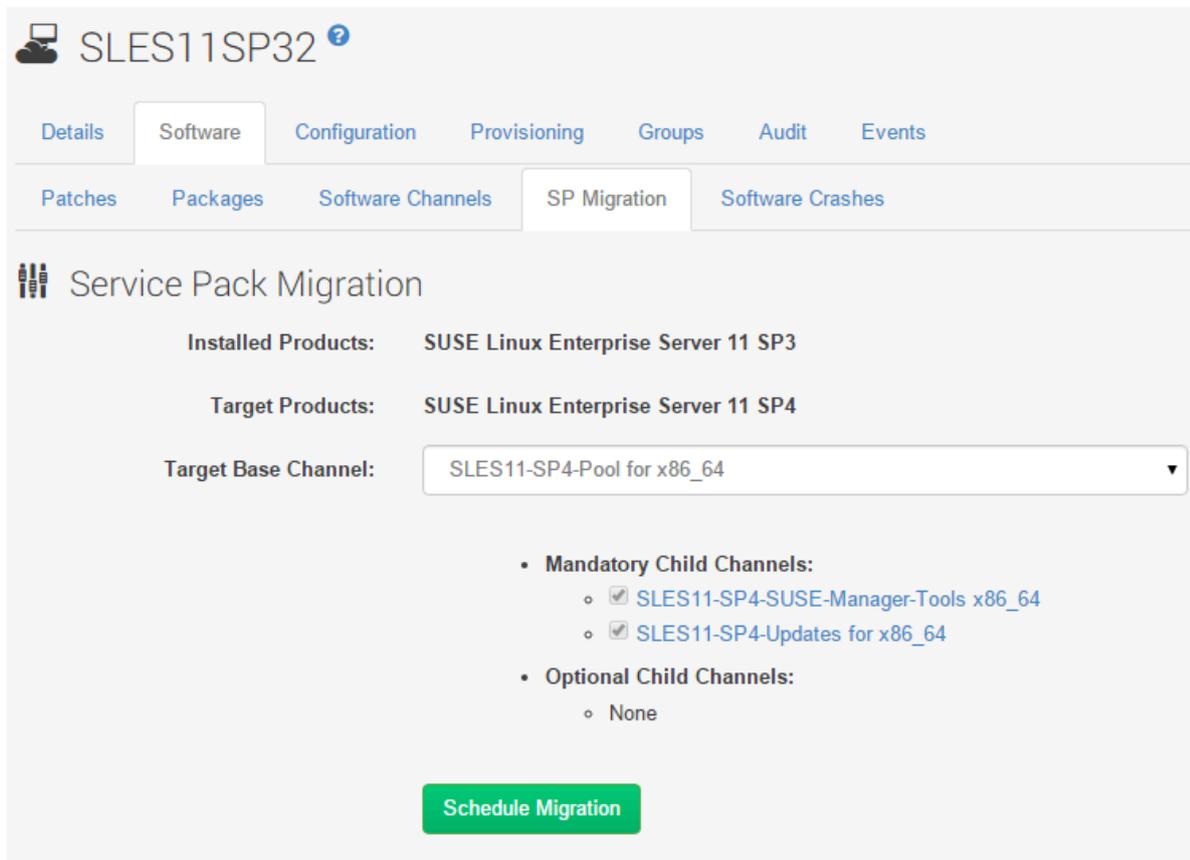See the following screenshot:

FIGURE 2: SERVICE PACK MIGRATION

To allow usage of the patch lifecycle processes described here **and** to use the functionality of service pack migrations, it is best to create a full clone-set of the SUSE Linux Enterprise Server product versions in use at your company. This is sometimes called a clone-tree, and can be easily created using the `spacecmd` command of the same name, `softwarechannel_clonetree`.

A full clone-tree will include the base/parent channel (normally called a pool) and all child channels that currently exist underneath it. For example, using the SUSE Linux Enterprise Server 11 SP3 Pool x86_64 channel as a source, you provide your own **prefix** to the clone-tree command and it will make a copy of the pool and all children adding the prefix to the beginning of all the channel names and labels.

See below for an image of the original SUSE Linux Enterprise Server 11 SP3 x86_64 channels and the cloned tree with prefix versions "Chameleon_Co-" next to it:

FIGURE 3: FULL SOFTWARE CHANNEL LIST FOR CHAMELEON CORPORATION

The command **`spacecmd softwarechannel_clonetree`** can be called directly by providing the source and prefix information directly–or interactively by calling the command from within the spacecmd shell itself.

Example:

```
spacecmd -u <SMadmin> -- softwarechannel_clonetree -s sles11-sp3-pool-x86_64 -p
 "my_company-"
```

🔖 Note: <SMadmin>

Make sure to replace <SMadmin> with your SUSE Manager webUI administrator user name!

This command creates a whole set of SUSE Linux Enterprise Server 11 SP3 x86_64 channels with a prefix of "my_company".

Repeat this process for each version of SUSE Linux Enterprise Server, and then–when using the Service Pack migration feature–you can choose **your** company version of the target version of SUSE Linux Enterprise Server, and choose the child channels appropriate for the landscape the host is in (as described below).

You can then modify the cloned updates channel by removing all patches/packages, then use the merge script process to merge your "Current Patch Set" into the mandatory updates channel. This way you can do service pack migrations, but still retain **your** currently promoted patch set. Otherwise, SP migrations will always update to whatever are the latest packages in the "mandatory" updates child channel.

Consider the following screenshot that shows the Q3 patch archive for SUSE Linux Enterprise Server 11 SP4 with 227 packages, the SUSE Updates channel for SP4 with 234 packages, and the Chameleon Corporation version of SP4 Updates with 0 (zero) packages:

Q3-2015 Patch Archive - 09-30-2015 - SLES11-SP3-Updates for x86_64	SUSE	3186	0
Q3-2015 Patch Archive - 09-30-2015 - SLES11-SP4-Updates for x86_64	SUSE	227	0
Chameleon_Co-SLES11-SP3-Pool for x86_64	SUSE	2862	1
Chameleon_Co-SLES11-SP4-Pool for x86_64	SUSE	2913	0
Chameleon_Co-SLES11-SP4-SUSE-Manager-Tools x86_64	SUSE	26	0
Chameleon_Co-SLES11-SP4-Updates for x86_64	SUSE	0	0
SLES11-SP3-Pool for x86_64	SUSE	2862	0
SLES11-SP4-Pool for x86_64	SUSE	2913	0
SLES11-SP4-SUSE-Manager-Tools x86_64	SUSE	26	0
SLES11-SP4-Updates for x86_64	SUSE	234	0

FIGURE 4: PARTIAL VIEW OF SOFTWARE CHANNEL LIST FOR SUSE AND CHAMELEON CORPORATION

In the upcoming sections we will describe the creation of custom "current updates" channels and how to use a merge script. For service pack migrations using your own set of promoted patches, merge a "Current" Patch Archive from a "production" channel into the "Updates" channel so it will be used for migration efforts–or simply leave the "Updates" channel always **empty** and add your Current Updates channel to the "Optional Child Channels"" selection.

With an **empty** "Updates" channel and a populated "Current Updates", you can be assured of a service pack migration working as expected and not patching the machine to a version of packages more current than whatever is deemed "production". The custom "Updates" channels and the merge script process will be described in detail in the coming sections.

Advanced Patch Lifecycle Management with SUSE Manager

Now create the "Current Updates" channels that managed hosts will subscribe to depending on their landscape, for example, "DEV - Current Updates - for SLES 11 SP3 x86_64", or "PROD - Current Updates - for SLES 12 x86_64". There will be a full set of landscape channels for each version of SUSE Linux Enterprise Server and they will all be empty to begin with. Create a "new" channel and then optionally "clone" that one to reduce the amount of typing needed:

1. From the "Manage Software Channels" submenu, click " + Create Channel".

2. Create the channel within one of the SUSE Linux Enterprise Pool Channels or, if you have your own clone set, within that pool.

3. Modify the channel to be Public.

4. Clone this new channel (without updates) into the remaining landscapes, each as a "Current Updates -" channel, and each with the appropriate landscape prefix (for example, "DEV - ", "TEST - ", "PROD - ", etc.).

5. You should end up with a full set of landscapes for each SUSE Linux Enterprise Server version you are going to manage. Keep in mind for the SUSE Linux Enterprise Server 11 SP1 and SP2 channels this will result in a full set x2 (times 2)–or even x4 (times 4) with LTSS. However, SP1 and SP2 Update channels are not being updated any longer, so they could be subscribed as static channels now. (See the note about LTSS in chapter *Section 2.3, "Patch Promotion Cycle"*.)

FIGURE 5: EXAMPLE SCREENSHOT OF "CURRENT UPDATES" CHANNELS

"Exception" Channels:

Now create the two exception channels for each version of SUSE Linux Enterprise Server being managed. While it may be possible to create a mixed channel that is unique to the architecture (like the archive channel base), tracking which distribution a patch exception has come from can get cumbersome. If you mix patches/packages from different versions of SUSE Linux Enterprise Server, it is more difficult to track, resolve, and then reintroduce these patches back into the patch workflow. It is much easier to place them into an exceptions channel specifically for the version of SUSE Linux Enterprise Server the patch came from. Create the "Patch Exception" and "Security ASAP Exception" channels as follows:

1. From the "Manage Software Channels" submenu, click " + Create Channel".

2. Create the "Patch Exceptions - DO NOT SUBSCRIBE" channel within one of the SUSE Linux Enterprise Server pool channels or, if you have your own clone set, within that pool. This channel will reside next to your landscape channels.

3. Modify the channel to be Public.

Advanced Patch Lifecycle Management with SUSE Manager

4. Create the "Security ASAP Exception" channel as above–it too will reside next to your landscape and patch exceptions channels.

5. For SUSE Linux Enterprise Server 11 SP1 and SP2 keep in mind: they reside in the same SP1 Pool base channel, so there will be exceptions and security exceptions channels for each service pack.

Chameleon_Co-SLES11-SP3-Pool for x86_64	2862
⌙ Chameleon_Co-SLES11-SP3-SUSE-Manager-Tools x86_64	104
⌙ Chameleon_Co-SLES11-SP3-Updates for x86_64	2582
⌙ DEV - Chameleon_Co - Current Updates - SLES11-SP3-Updates for x86_64	0
⌙ Patch Exceptions - (DO NOT SUBSCRIBE) Chameleon_Co - Current Updates - SLES11-SP3-Updates for x86_64	0
⌙ PROD - Chameleon_Co - Current Updates - SLES11-SP3-Updates for x86_64	3186
⌙ QA - Chameleon_Co - Current Updates - SLES11-SP3-Updates for x86_64	0
⌙ Security ASAP Exceptions - Chameleon_Co - Current Updates - SLES11-SP3-Updates for x86_64	0

FIGURE 6: **EXAMPLE SCREENSHOT OF "EXCEPTIONS" CHANNELS**

2.7   Activation Keys / Bootstraps

Typically you will have already created a set of Activation Keys assigned to a specific version of SUSE Linux Enterprise Server that your hosts are installed with. There may be other characteristics assigned in your keys like System Groups, assorted Child Channels, Software Packages, etc. You can choose to modify your existing Activation Keys and reuse them, or you can create new keys for this implementation. This example assumes a new key.

1. Within the "Systems" menu -> click the "Activation Keys" submenu from the left panel.

2. Click "+ Create Key" in the upper right.

3. Create the Activation Key within this Organization (remember to log in to other Organizations to create keys there if needed). Assign the appropriate entitlements, software packages, and configuration channels as appropriate.

4. Name the Activation Key according to the **Landscape** host you will register to with this key.

Example:

FIGURE 7: CREATE ACTIVATION KEY SCREENSHOT

5. **Important:** Click on the Child Channels Tab. Now you can assign the appropriate Current Updates channel depending on which **Landscape** this key is for.

Example:

FIGURE 8: CHILD CHANNELS SCREENSHOT

Advanced Patch Lifecycle Management with SUSE Manager

6. Select the appropriate SUSE Manager Tools channel and the other needed channels to fill out the requirements for the version of SUSE Linux Enterprise Server in question.

7. Note that the **Security ASAP Exceptions** channel **is** selected as an active child channel, and the **Patch Exceptions (DO NOT SUBSCRIBE)** channel is **not** selected. This is important, because any patch/package copied into the Patch Exceptions channel should remain invisible to hosts while patches/packages copied to the Security ASAP Exceptions channel should always be immediately visible to a host.

The Activation Key is called by a bootstrap script–often there is a one-to-one relationship between a registration bootstrap script and an activation key. Whether new activation keys were created or existing ones were modified to point to the landscape specific "current updates" channels, there should be a bootstrap script that calls that activation key.

All new hosts will use these bootstraps to register and become managed by SUSE Manager. One of the key benefits is that they should never need to change their channel assignments, as the host will now be assigned to its proper landscape update channels and will receive new updates as they are promoted each patch cycle. We will discuss promotion in the next section *Section 2.8, "Scripts"*.

2.8 Scripts

Patch/Package Merge Script:

The "Patch/Package Merge" script is used to **promote** patches (and relevant packages) from one channel to another. It is used at first to merge content from an Archive channel into the first stage (**landscape**), for example DEV. It is then used to promote content from one stage to the next until the content reaches the final stage. This process is then repeated for each version of SUSE Linux Enterprise Server (like 11 SP1, SP2, etc.) and each corporate environment (like STORE, CORP, etc.).

Finally, the entire process is repeated for each patch calendar cycle (quarterly/monthly/etc.).

Channel Archive Creation Script / Channel Archive Sources List Configuration File:

This script and its associated "sources list" configuration file are called either manually or by a crontab entry. The script uses the sources list file to determine which versions of SUSE Linux Enterprise Server and which architectures to create archives for. As previously described, the Archive channels are created as point-in-time clones of the various SUSE Linux Enterprise Server Updates channels.

To have the correct Archive channels created, simply add lines to the sources file. Each line has three fields, the first being the architecture, the second being the source channel, and the third being target channel text. The script parses the source list and constructs the target parent channel for the archives by architecture, and then constructs the target channel and label, etc.

The script calls the `spacecmd` functions `softwarechannel_clone` and `softwarechannel_setorgaccess` for each entry in the sources list. It calculates the current calendar quarter, creates the archives and makes these archives publicly visible. See the Appendix for the entire script.

2.9   Crontab Entries / Automation

Schedule-based Trigger for Archive Creation Script (Quarter-End)

A decision needs to be made whether a calendar quarter is going to get created (looking backward) at the end of a particular month, or at the beginning of a month (one day past the final day of a quarter). Since the Archive channels are getting created by cloning the SUSE Updates channels of a particular SLES version, the decision can affect the content.

Either way is fine as it will be consistent from quarter to quarter. For the purposes of explaining the cron syntax, we will opt to show the more difficult of the two: how to call the archive script at the end of the quarter. This is more difficult simply because it requires two cron entries to handle both the 30th of the month (for June and September) and the 31st of the month (for March and December).

The crontab entries will point to the archive creation script and execute in months 3, 6, 9, and 12. Here are what the cron entries look like:

```
0 0 30 6,9 * /path/to/the/archive/script
```

```
0 0 31 3,12 * /path/to/the/archive/script
```

Advanced Patch Lifecycle Management with SUSE Manager

The path will point to the actual archive-channel-create-script.sh (or similar). This could be placed in /usr/sbin/ and must be made *executable* (chmod + x).

Also it is very important to note that this script is calling `spacecmd` commands from within a Bash shell script. Spacecmd is itself a python-based shell that can be invoked–allowing the use of an extensive set of commands **within** the shell, or from outside the shell as a mediated command interpreter.

The invocation of spacecmd stores a **.spacecmd** directory in the invoking user`s home directory. Within the .spacecmd directory is a configuration file that can store credential sets that can be leveraged to avoid being prompted at spacecmd execution. This can be used to store credentials so the cron job will execute and not pause waiting for authentication credentials to be passed.

The crontab entries will point to the archive creation script and execute in months 3, 6, 9, and 12. Here are what the cron entries look like:

```
[spacecmd]
server=localhost
username=admin
password=suse1234
nossl=0
```

Storing these credentials will allow `spacecmd` operations from the SUSE Manager's root user to be executed without the need to pass user name or password–allowing the cron job for archive creation to run without failure at the appropriate time.

3 Usage Workflows

This guide describes an approach to patch lifecycle management that leverages SUSE Manager to deliver the following benefits:

- Automated creation of Patch Archive Channels:

  - These channels can be created on a quarterly (or more frequent) basis and allow an organization to create test environments based on a historical set of patches (for example, the creation of a lab using available patches from two calendar quarters ago).

- Leverage a static set of "current" update channels so host subscriptions do not need to change:

  - Using the API/scripts, updates can be merged from a patch "archive" into a subscribed host channel removing the need to constantly clone and re-clone channels and modify host subscriptions.

  - Multi-landscape environments can use a promotion process to merge updates through each phase during testing/validation of patch sets.

- Exception handling for patches that need to be excluded from a patch roll out:

  - The creation of a "patch exceptions" channel and a process for copying patches/packages into that channel (and then removing them from an updates channel) allows for tracking of patch exceptions. This channel should be *excluded* from any hosts channel subscriptions, thereby keeping all patch/package content from being visible or available for installation.

  - Patch exceptions "processes" must be developed to track remediation of all exceptions to avoid future complications from managing an ever-growing bucket of patches.

- Security ASAP Exceptions handling for patches that need to roll out with a higher priority:

  - The security ASAP exception channel should always be subscribed to (unlike the "patch exceptions" channel). Subscription to this channel allows a host to obtain patches/packages added in an ad hoc manner, for example a security patch deemed important enough to deploy outside of a normal patch schedule. This could be copied from a new archive or directly from one of the SUSE (vendor) updates channels.

3.1   Example Process Workflows

Here are some suggested workflows for the patch promotion and exception processes. Keep in mind that these can be modified to fit more closely to a particular operational group's existing set of processes.

Workflow 1: Patch Promotion Process

The patch promotion process follows the following sample steps.

TABLE 1: PATCH PROMOTION PROCESS

Action	Process	Notes
Review	Refer to the existing calendar to determine the start date for a patch roll out.	Start dates typically happen when a new quarter begins, but this depends on your frequency and established periods for roll outs.

Action	Process	Notes
Select	Identify the SUSE Manager Organization and the environment targets for the patch roll out.	In the scenario here, this could be CORP or STORE or NPE, and depending on the use of SUSE Manager Organizations an administrator would need to log in to the Organization to see the hosts.
Select	Choose the SUSE Linux Enterprise Server version(s) that will be patched.	Patch promotions are done for each SUSE Linux Enterprise Server version. Each version has its own set of landscape channels that patches get promoted through (DEV, TEST, QA, PROD, etc.).
Merge	Merge the current archive into the initial landscape (or landscape of choice).	Using the "Merge Script" utility: select the **source** channel (**archive** in this case) and **target** channel (DEV - Current to begin a series).
Deploy	Deploy patches from the current landscape to the subscribed hosts.	When patches have been merged, the status of hosts subscribed to the "current" channel will show which patches are now available. Issue the appropriate patch commands to deploy these patches.
Test	Coordinate testing for the hosts that have received the latest deployed patches, or notify the appropriate teams/LOB to start their evaluation.	After the patches have been deployed, a period of testing or review should start–in order to validate the success of the patch deployment. Coordination should occur with business partners to establish success of deployment.
Evaluate	Evaluate the results of the previous merge/deploy and proceed with the	Continue to merge and deploy patch sets into each landscape until the fi-

Action	Process	Notes
	next landscape. If final landscape, report completion.	nal one. Report completion–all hosts should show "green" status.

Workflow 2: Patch Exception Process

The patch exception process follows the following sample steps. Again, keep in mind that these can be modified to fit more closely to a particular operational group's existing set of processes. Consider the following table:

TABLE 2: PATCH EXCEPTION PROCESS

Action	Process	Notes
Identify	Locate the patch you want to identify as an "exception" for potential removal from patch set.	Find the patch using the search tools in SUSE Manager, or locate the patch in a specific Archive or Landscape Channel.
Copy	Using the "Manage Channels" function of SUSE Manager: select the **exception** channel for a specific version of SUSE Linux Enterprise Server and add the identified patch into it from the source (from previous step).	The source of the patch (channel) and the target of the patch (exceptions channel) are important. This step may be repeated for different versions of SUSE Linux Enterprise Server. Potential sources for the patch can be an: a. Archive Channel b. Landscape Channel c. SUSE Updates Channel
Remove	Using the "Manage Channels" function of SUSE Manager: *list* the patches to find the patch that was "added" in the previous step. Find it	Finding the patch and then removing it from a Landscape Channel keeps the subscribed hosts from seeing the patch–and reporting its applicability.

Action	Process	Notes
	and then **remove** it from the current phase (or phases).	
Track	Submit any tickets or commence any established process to track this patch exception.	The goal of tracking is to remain aware that there is an exception and attempt to re-mediate, allowing the patch to be reintroduced into a patch deployment cycle.
Remediate	Work to re-mediate the exception.	When a fix has been identified/created for a given exception, the patch can be reintroduced into a deployment workflow.
Note	Exceptions should only last for a **single** deployment cycle. Keep in mind the next archive will also contain this same patch–which is a **good thing**.	Exceptions should always be a temporary condition. Work should always be done to fix the reasons a patch cannot roll out– compliance can be at risk while an exception exists.

Workflow 3: Security Patch ASAP Process

The security exception process differs from the previous patch exception process in that a patch is now being **added** to a patch roll out cycle–likely in the middle of a current (inprogress) roll out. Another process table and some screenshots from the SUSE Manager interface are included here to provide further clarity.

Review the following table:

TABLE 3: SECURITY PATCH ASAP PROCESS

Action	Process	Notes
Identify	Locate the patch you want to identify as a "security exception"–for potential **addition to** a particular landscape or patch set.	Find the patch using the search tools in SUSE Manager, or locate the patch in a specific Updates Channel

Action	Process	Notes
		for a specific version of SUSE Linux Enterprise Server.
Copy	Using the "Manage Channels" function of SUSE Manager, select the Security ASAP Exception Channel for a specific version of SUSE Linux Enterprise Server and *add* the identified patch to it from the source (likely the SUSE Updates Channel from the previous step).	The source of the security patch (SUSE Updates Channel) and the target of the patch (Security Exceptions ASAP Channel) is key. This step may be repeated for different versions of SLES. Potential source for the patch will likely be an: a. Archive Channel b. Landscape Channel c. SUSE Updates Channel
Deploy	Deploy the Security Exception patch(es) into the selected landscape to the subscribed hosts.	When the Security patches have been added to a Landscape Channel, the status of hosts subscribed to the "current" channel will show patches are now available. Issue the appropriate patch commands to deploy these patches.
Track	Submit any tickets or commence any established process to track this Security Patch exception.	The goal of tracking is to remain aware that a Security Patch exception has been deployed. This patch will become part of a normal archive during the next quarterly automation creation.
Report	Security Exceptions normally occur to handle a compliance concern. When deployed, a compliance (Audit) report can be generated.	Security Exceptions are handled as an escalation to the normal schedule. They will, as a matter of normal operation, be part of the next Archive and would be deployed as part of the default schedule– and as

Action	Process	Notes
		a corollary, the next normal roll out would not need to include this patch (as it will already be deployed).

This screenshot shows the function of adding a patch to the Security ASAP Exceptions Channel–specifically the SLES 11 SP1 x86_64 version:
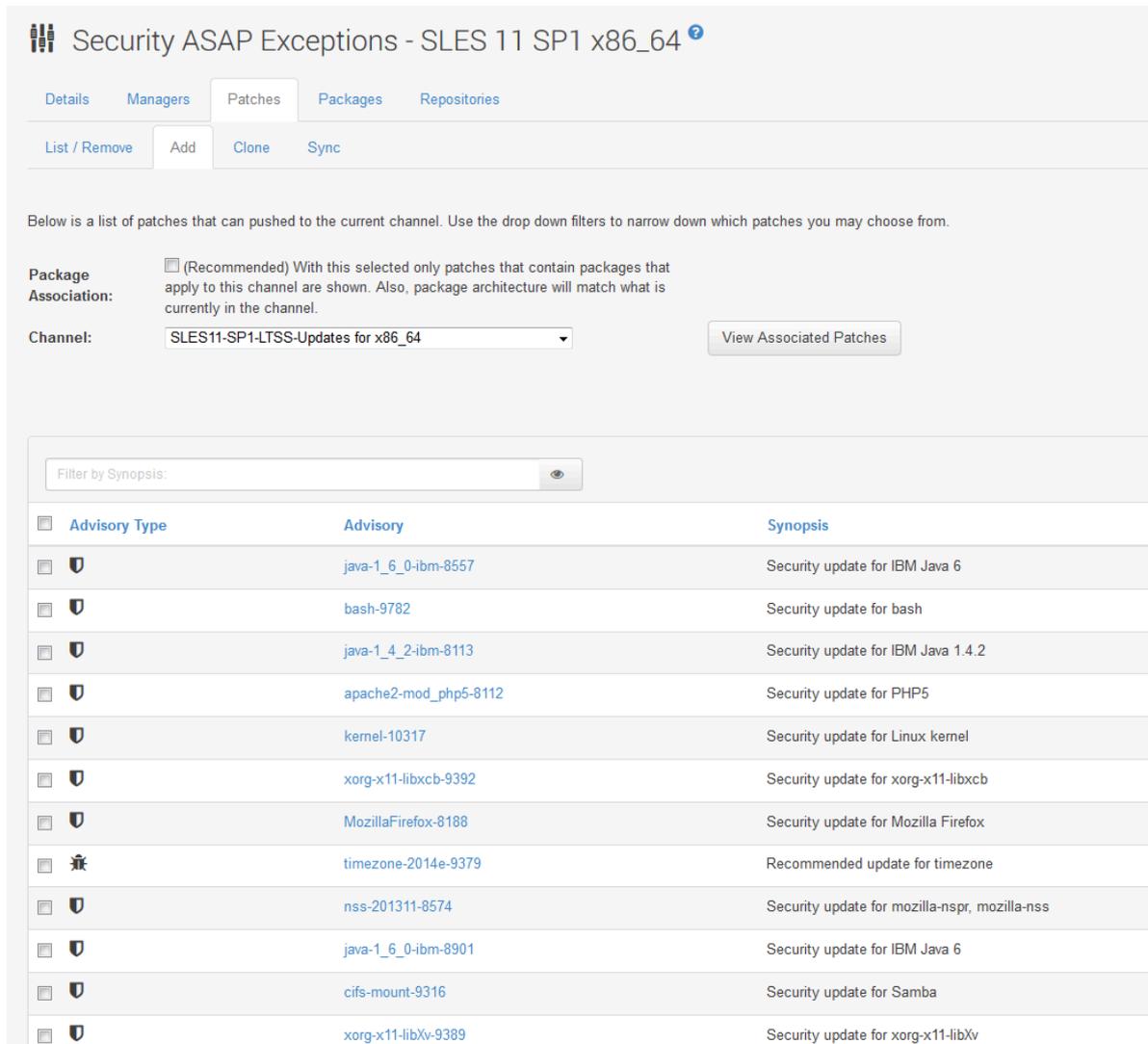
This screenshot provides a view of the **List/Remove** function of a specific channel (in this case, "DEV - Current - SLES11-SP1-LTSS-Updates for x86_64"), and would be the interface where a patch would be removed from a channel and keep it from being visible and potentially deployed as part of an exception process.

Advanced Patch Lifecycle Management with SUSE Manager
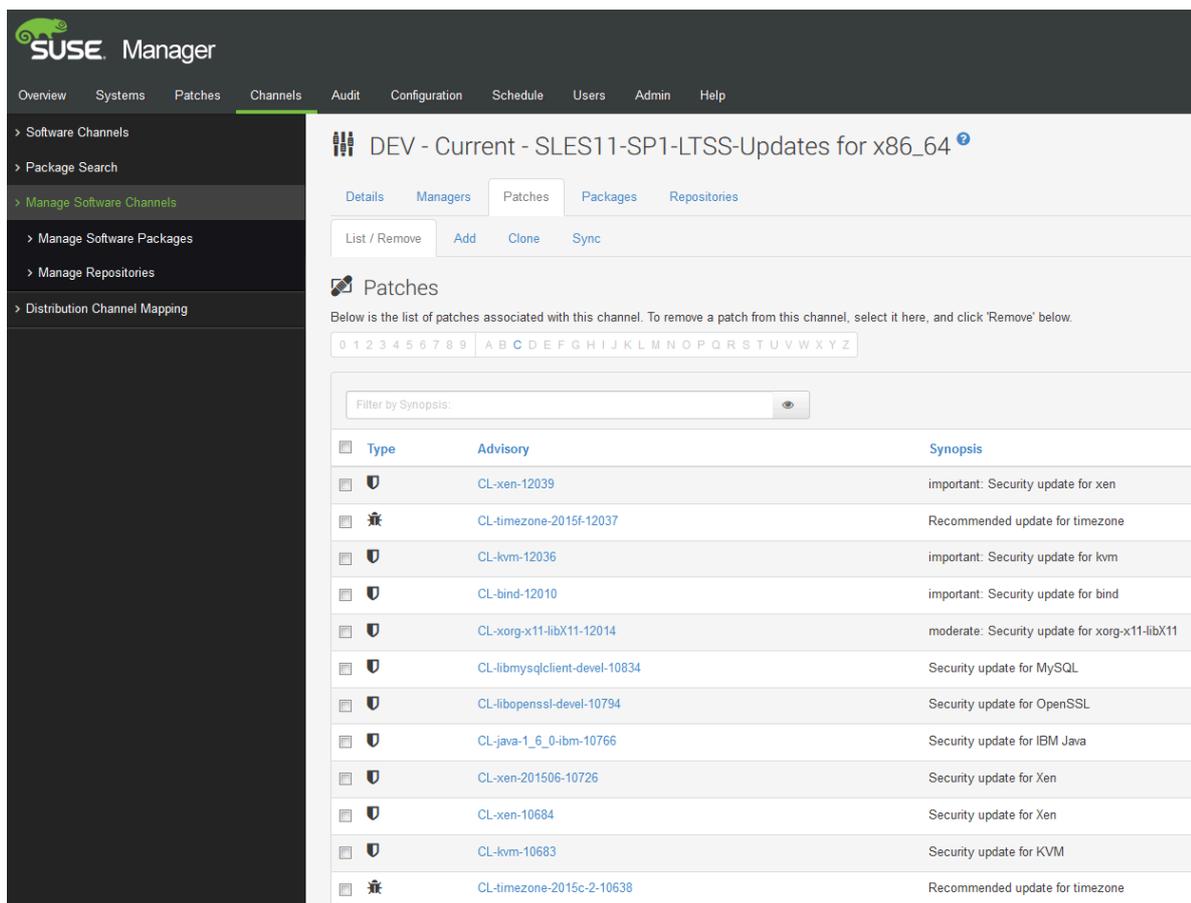
FIGURE 10: **ADDING A PATCH TO THE SECURITY ASAP EXCEPTIONS CHANNEL**

4 Appendix

4.1 Spacewalk-Clone-By-Date Configuration File Sample

This sample is stored as a text file. It is used with the **spacewalk-clone-by-date** utility by using the **-c** flag to indicate an input configuration file:

```
spacewalk-clone-by-date -c Q3-2015-sles11sp3-x86_64-archive.conf
```

Note that it specifies the base channel source **sles11-sp3-pool-x86_64** and the target base channel (destination) **cc_patch_archives_channel_64bit** and uses the directive of **existing-parent-do-not-modify : true.** This tells the utility that a child channel from one base will be cloned

Advanced Patch Lifecycle Management with SUSE Manager

into a completely different base channel. Note that your channel names might be different; this means this example may need modifications to work in your case. Example: **Q3-2015-sles11sp3-x86_64-archive.conf** contents below.

```
{
 "username":"SMadmin",
 "to_date": "2015-09-30",
 "skip_depsolve":false,
 "security_only":false,
 "use_update_date":false,
 "no_errata_sync":false,
 "dry_run":false,
 "channels":[
       {
         "sles11-sp3-pool-x86_64": {
            "label": "cc_patch_archives_channel_64bit",
            "existing-parent-do-not-modify": true
         },
         "sles11-sp3-updates-x86_64": {
            "label": "09_30_2015_q3_archive-sles11-sp3-updates-x86_64",
            "name": "Q3-2015 Patch Archive - 09-30-2015 - SLES11-SP3-Updates for x86_64",
            "summary": "Q3 - 2015 - Patch Archive Set (9-30-2015) - SUSE Linux Enterprise
Server 11 SP3 x86_64",
            "description": "Q3 - 2015 - Patch Archive Set (9-30-2015) - SUSE Linux
Enterprise Server 11 SP3 x86_64"
         }
       }
       ]
}
```

Note: Review Channel Patch Content

When using the spacewalk-clone-by-date utility, review the channel patch content after creation to ensure that the channel contains the patches for the appropriate end date. Sometimes extra patches are included in the channel that are in the wrong time period and can be deleted manually.

4.2   Sample Patch/Package Merge Script Using Python

This script has a defined SUSE Manager server URL set in the `MANAGER_URL`. The script prompts for a SUSE Manager Administrator ID and password. It then asks for a source and target channel. The script will then confirm the channels and ask if it should continue. An affirmative response (Y) will then allow the script to merge the patches and packages from the source channel into the target channel.

```python
#!/usr/bin/python
import xmlrpclib
import sys
import getpass

MANAGER_URL = "https://suma01.chameleoncorp.com/rpc/api"
MANAGER_LOGIN = raw_input("Please Enter the SUSE Manager Login Name: ")
MANAGER_PASSWORD = getpass.getpass("Please Enter the Password: ")

MERGE_SOURCE = raw_input("Enter the SOURCE channel label to Merge FROM: ")
MERGE_TARGET = raw_input("Enter the TARGET channel label to Merge INTO: ")

print("This tool is going to take all packages and errata from the SOURCE")
print("Channel : " + MERGE_SOURCE)
print("and merge it into the TARGET ")
print("Channel : " + MERGE_TARGET)

def query_yes_no(question, default="yes"):
    """Ask a yes/no question via raw_input() and return their answer.

    "question" is a string that is presented to the user.
    "default" is the presumed answer if the user just hits <Enter>.
        It must be "yes" (the default), "no" or None (meaning
        an answer is required of the user).

        The "answer" return value is True for "yes" or False for "no".
        """
        valid = {"yes": True, "y": True, "ye": True,
                 "no": False, "n": False}
        if default is None:
            prompt = " [y/n] "
        elif default == "yes":
            prompt = " [Y/n] "
        elif default == "no":
            prompt = " [y/N] "
        else:
            raise ValueError("invalid default answer: '%s'" % default)
```

```
        while True:
            sys.stdout.write(question + prompt)
            choice = raw_input().lower()
            if default is not None and choice == '':
                return valid[default]
            elif choice in valid:
                return valid[choice]
            else:
                sys.stdout.write("Please respond with 'yes' or 'no' "
                                 "(or 'y' or 'n').\n")

query_yes_no("Is this information correct?")

client = xmlrpclib.Server(MANAGER_URL, verbose=0)
key = client.auth.login(MANAGER_LOGIN, MANAGER_PASSWORD)

client.channel.software.mergePackages(key, MERGE_SOURCE, MERGE_TARGET)
client.channel.software.mergeErrata(key, MERGE_SOURCE, MERGE_TARGET)

client.auth.logout(key)
```

4.3   Sample Crontab Entries for Automation of Archive Channel Creation

Below are some sample cron entries that can be used to run the **Archive Channel Creation Script** resulting in channel creation for each quarter. This example has two entries to clone all patches received up until the end of a quarter. A single entry could be used to create it at the beginning of a quarter, but care must be taken to modify the calculations in the **Archive Script** to account for that. Currently the example **Archive Channel Creation Script** below figures out what quarter it is by looking at the current date. Running the script in the 4th quarter will name the archive created as a 4th quarter archive.

Cron entries for end of quarter: (for example 30th of months June and September and 31st of months March and December):

```
0 0 30 6,9 * /path/to/the/archive/script
```

```
0 0 31 3,12 * /path/to/the/archive/script
```

4.4 Sample Archive Channel Creation Script for SUSE Manager 2.1 and 3.0

The script below works with the **Archive Channel Sources** file (see *Section 4.6, "Sample Archive Channel Sources List File"*) to create quarterly archives of the SUSE Updates channels. This example script is written in **Bash** but calls `spacecmd` to accomplish the cloning and to set the new archive to be public (accessible to other organizations).

The script takes some time to run. The clone command finishes quickly, but the actual cloned channel still takes some time to settle down before the `org-access` command can access and finish. If you want to test this manually you must be patient. A normal crontab-type run of the script will finish in due time.

```
#!/bin/bash


    ###################################################################
    #    SUMA Archive Channel Creation Script - Called from Cron      #
    #                                                                 #
    #    This script creates quarterly archives of SUSE Manager       #
    #    channels from SUSE Updates channels. It takes a list         #
    #    of source channels from the archive-sources.lst file         #
    #    that should be located in the same directory as this         #
    #    script. Each entry in that file will be used as a            #
    #    source channel to create an archive for patches/updates      #
    #    in the appropriate archive channel.                          #
    #                                                                 #
    # REQUIRES:                                                       #
    #               1. cron entries for each quarter :                #
    #      e.g. 30th of months June and Sept. and 31st of months March #
    #        and December:                                            #
    #               0 0 30 6,9 * /path/to/this/script                 #
    #               0 0 31 3,12 * /path/to/this/script                #
    #                                                                 #
    #               2. archive-sources.lst :                          #
    #      A list of the architecture, the source updates channel     #
    #       for each distro and the suffix of the target channel      #
    #       version and architecture (1 per line - no line-feed at EOF) #
    #   Example:                                                      #
    #  S390x,sles11-sp3-updates-s390x,SLES11-SP3-Updates for s390x    #
    #  ppc64,sles11-sp4-updates-ppc64,SLES11-SP4-Updates for PPC      #
    #  x86_64,sles11-sp3-updates-x86_64,SLES11-SP3-Updates for x86_64 #
    #         etc.                                                    #
    #                                                                 #
    ###################################################################
    #                                                                 #
```

```
        #        Created by - Jeff Price, SUSE Consulting - 2015        #
        #                                                               #
        ################################################################

## date strings
month=`date +%m`
year=`date +%Y`
fdate=`date +%m-%d-%Y`
## set quarter
if [ $month -le 3 ]
then
    quar=1
elif [ $month -gt 3 ] && [ $month -lt 7 ]
then
    quar=2
elif [ $month -gt 6 ] && [ $month -lt 10 ]
then
    quar=3
elif [ $month -gt 9 ]
then
    quar=4
fi

## Create archives using source list

while read line
do
        arch=`echo "$line" | awk -F, '{print $1}'`
        src_ch=`echo "$line" | awk -F, '{print $2}'`
        trg_ch=`echo "$line" | awk -F, '{print $3}'`
## set archive channel

target_parent=$arch"-patch-archives-channel"
source_channel=$src_ch
target_channel_name="Q"$quar" "$year" - "$fdate" - Archive of "$trg_ch
target_summary="Q"$quar"-"$year" Archive Set "$trg_ch
target_channel_label="q"$quar"-"$year"-archive-"$src_ch

## Debug Output
echo "Architecture: " $arch
echo "Source Channel: "$src_ch
echo "Target Channel Archive Suffix: "$trg_ch
echo "Target Archive Parent Channel: "$target_parent
echo "Source Channel (again): "$source_channel
echo "Target Channel Name: "$target_channel_name
lctn=`echo $target_channel_name|tr '[:upper:]' '[:lower:]'`
echo "lowercase target name: "$lctn
```

```
echo "Target Channel Label: "$target_channel_label
echo "Target Channel Summary and Description: "$target_summary

/usr/bin/spacecmd -d -- softwarechannel_clone -s "'$src_ch'" -n
"'$target_channel_name'" -l "'$target_channel_label'" -p "'$target_parent'" - g
/usr/bin/spacecmd -d -- softwarechannel_setorgaccess
"'$target_channel_label'" -e
done < ./archive-sources.lst
```

4.5 Sample Archive Channel Creation Script for SUSE Manager 3.1 and 3.2

The script below works with the **Archive Channel Sources** file (see *Section 4.6, "Sample Archive Channel Sources List File"*) to create quarterly archives of the SUSE Updates channels. This example script is written in **Bash** but calls `spacecmd` to accomplish the cloning and to set the new archive to be public (accessible to other organizations).

The script takes some time to run. The clone command finishes quickly, but the actual cloned channel still takes some time to settle down before the `org-access` command can access and finish. If you want to test this manually you must be patient. A normal crontab-type run of the script will finish in due time.

```
#!/bin/bash

    ####################################################################
    #    SUMA Archive Channel Creation Script - Called from Cron      #
    #                                                                 #
    #    This script creates quarterly archives of SUSE Manager      #
    #    channels from SUSE Updates channels. It takes a list        #
    #    of source channels from the archive-sources.lst file        #
    #    that should be located in the same directory as this        #
    #    script. Each entry in that file will be used as a           #
    #    source channel to create an archive for patches/updates     #
    #    in the appropriate archive channel.                         #
    #                                                                 #
    # REQUIRES:                                                       #
    #               1. cron entries for each quarter :               #
    #     e.g. 30th of months June and Sept. and 31st of months March #
    #       and December:                                            #
    #               0 0 30 6,9 * /path/to/this/script                #
    #               0 0 31 3,12 * /path/to/this/script               #
    #                                                                 #
    #               2. archive-sources.lst :                         #
```

Advanced Patch Lifecycle Management with SUSE Manager

```
       #     A list of the architecture, the source updates channel    #
       #     for each distro and the suffix of the target channel      #
       #     version and architecture (1 per line - no line-feed at EOF) #
       #   Example:                                                     #
       # S390x,sles11-sp3-updates-s390x,SLES11-SP3-Updates for s390x   #
       # ppc64,sles11-sp4-updates-ppc64,SLES11-SP4-Updates for PPC     #
       # x86_64,sles11-sp3-updates-x86_64,SLES11-SP3-Updates for x86_64 #
       #        etc.                                                    #
       #                                                                #
       ##################################################################
       #                                                                #
       #      Created by - Jeff Price, SUSE Consulting - 2015           #
       #      Updated for SUMA 3.1 & 3.2 - Levin Forrest,        #
       #      Lenovo Consulting - 2018                             #
       #                                                                #
       ##################################################################

## date strings
month=`date +%m`
year=`date +%Y`
fdate=`date +%m-%d-%Y`
## set quarter
if [ $month -le 3 ]
then
   quar=1
elif [ $month -gt 3 ] && [ $month -lt 7 ]
then
   quar=2
elif [ $month -gt 6 ] && [ $month -lt 10 ]
then
   quar=3
elif [ $month -gt 9 ]
then
   quar=4
fi

## Create archives using source list

while read line
do
       arch=`echo "$line" | awk -F, '{print $1}'`
       src_ch=`echo "$line" | awk -F, '{print $2}'`
       trg_ch=`echo "$line" | awk -F, '{print $3}'`
## set archive channel
## Note the target_parent requires this EXACT channel already created
target_parent=$arch"-patch-archives-channel"
source_channel=$src_ch
```

```
target_channel_name="Q"$quar" "$year" - "$fdate" - Archive of "$trg_ch
target_summary="Q"$quar"-"$year" Archive Set "$trg_ch
target_channel_label="q"$quar"-"$year"-archive-"$src_ch


## Debug Output
echo "Architecture: " $arch
echo "Source Channel: "$src_ch
echo "Target Channel Archive Suffix: "$trg_ch
echo "Target Archive Parent Channel: "$target_parent
echo "Source Channel (again): "$source_channel
echo "Target Channel Name: "$target_channel_name
lctn=`echo $target_channel_name|tr '[:upper:]' '[:lower:]'`
echo "lowercase target name: "$lctn
echo "Target Channel Label: "$target_channel_label
echo "Target Channel Summary and Description: "$target_summary


## removed double quotes from $src_ch to allow spacecmd to parse correctly
## removed single quotes from $target_channel_label and $target_parent to allow spacecmd
 to parse correctly
## only $target_channel_name should have single quotes as this variable contains spaces
/usr/bin/spacecmd -d -- softwarechannel_clone -s $src_ch -n "'$target_channel_name'" -l
 "$target_channel_label" -p "$target_parent" -g
## removed double quotes from $target_channel_label to allow spacecmd to parse correctly
/usr/bin/spacecmd -d -- softwarechannel_setorgaccess $target_channel_label -e
## added full directory structure to archive-sources location to allow call from crontab
done < /usr/sbin/archive-sources.lst
```

4.6   Sample Archive Channel Sources List File

This file is called and used by the Archive Channel Creation script located above in the previous section (see *Section 4.4, "Sample Archive Channel Creation Script for SUSE Manager 2.1 and 3.0"*). Since the function "readline" is used, there should only be lines with data in the file. Any blank lines will be sourced as data and will cause errors with the Archive Channel Creation script above.

The first field is **architecture**, which defines the source parent archive channel. This is where the new archive will be placed as a child channel. The second field is the **source channel label**, which is where the patches/packages are coming FROM. The last field is used for **target channel naming**. This field will be part of the text that makes up the channel name and part of the channel summary/description fields.

```
s390x,sles11-sp3-updates-s390x,SLES11-SP3-Updates for s390x
x86_64,sles11-sp4-updates-x86_64,SLES11-SP4-Updates for x86_64
i586,sles11-sp3-updates-x86_64,SLES11-SP3-Updates for x86_64
```

4.7   Spacecmd Automatic Authentication

The archive script example here is written in Bash and calls the spacecmd shell. Normal operation of spacecmd and the SUSE Manager XMLRPC api requires authentication credentials. In order for the archive script to run successfully from cron, it would require some way of passing appropriate user credentials when required. Luckily there is a way to store user credential pairs.

Upon the first invocation of the spacecmd utility/shell, a directory is created in the home directory of the user who calls spacecmd. The directory is called .spacecmd (for example /root/.spacecmd/). Within this directory is a file called "config", and within this file you can store parameters that will be used when the user subsequently calls spacecmd. The parameters are as follows (substitute your user name and password):

```
[spacecmd]
server=localhost
username=admin
password=susemgr
nossl=0
```

When those credentials are stored, you should be able to invoke the spacecmd shell or call spacecmd commands from the command line without being prompted for a user name or password.

5 Legal notice

Copyright ©2006-2024 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled "GNU Free Documentation License".

SUSE, the SUSE logo and YaST are registered trademarks of SUSE LLC in the United States and other countries. For SUSE trademarks, see http://www.suse.com/company/legal/ ↗. Linux is a registered trademark of Linus Torvalds. All other names or trademarks mentioned in this document may be trademarks or registered trademarks of their respective owners.

Documents published as part of the **SUSE Best Practices** series have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

Below we draw your attention to the license under which the articles are published.

GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/ ↗.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
 Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts". line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.