Maintenance/SLE11 SP2 MM FAQ

From Novell Wiki

Contents

What is the intention of the SLE11/"New" Maintenance Model?

The high level goals are:

- Smaller, less heavy Service Packs, which are easier to consume
- A new online migration method that tries to provide a "minimal" upgrade versus the previous "full" upgrade that remains available as well. Also, the channel layout scheme has been adjusted mainly to match the requirements of the migration method.
- Feature updates also between Service Packs, provided via Maintenance

Which products participate in it?

- SLES 11 SP1/SP2, also SLES for VMware and SLES for SAP
- SLED 11 SP1/SP2
- SDK 11 SP1/SP2

All other products use individual update policies that are sometimes similar (e.g. for SLE11 HA), but they continue to use the traditional channel layout well known from SLE10.

How can I help with keeping Service packs small?

The primary way to do that is to fix bugreports as early as possible, and submit bugfix-only updates via maintenance, instead of waiting for the SP3 development cycle to start.·

Any submission with fixes that are handled via maintenance and are released as an update will automatically enter the SP3 media and SP3 development without any extra effort on your side. it will reduce the number of package branches for each service pack, and in the end reduce your and our work in maintaining the package over a long period of time.

What is the new channel layout ?

Previously, a registered system got a "productname-SPx-Pool" and "productname-SPx-Updates" channel assigned for each installed product. For the SLE11/MM products (see above), starting with SP2 this has been changed to the following model:

- all -Update channels from previous SPs, down to SP1 are subscribed
- SP1-Pool is subscribed
- for all newer service packs, a SPx-Core channel is introduced.

So for example, a SP2 system has:

```
- SP1-Pool
- SP1-Updates
- SP2-Core
- SP2-Updates
```

subscribed. a SP3 system would get all of the above _and_ additionally SP3-Core and SP3-Updates.

The SP2-Core channel provides a subset of packages of the particular product that have been changed as part of SP2 development (aka it is the content of SUSE:SLE-11-SP2:GA binary packages filtered by product).

Packages that are not part of the SP2-Updates or SP2-Core channel are maintained via SP1-Updates channel, which means that they are also at the same time available for SP1 based products, as they share the identical update channel.

This means that anything submitted to SP1-Updates needs to be fully compatible both with SP1 and with SP2 at the same time, unless there a SP2-Updates specific version of the package is being maintained as well.

The complexities of the new channel model are explained in a bit more depth here: http://www.novell.com/support/kb/doc.php?id=7010225

What changes for the maintenance lifecycle with the SLE11/"New" Maintenance Model?

The executive summary: Nothing.

With the release of a new service pack, the old service pack is entering a 6 month phase where maintenance and security updates are provided equally for both the SPx product as well as the SPx-1 product.

After 6 months, the SPx-1 maintenance is discontinued and those package versions only valid for SPx-1 are no longer updated.

The only exception is "Long Term Service Pack Support (LTSS) for SLES", which means that as an exception, the maintenance or security team might request you to update the package with a critical fix for SLES packages on that particular SPx level for an additional 3 years.

What changes for a package maintainer/SUSE Developer with the SLE11/"New" Maintenance Model?

The executive summary is: There are no major differences to the way SLE10 handled maintenance for the packager side.

Packages that have been checked into **SUSE:SLE-11-SP2:GA** in the IBS are maintained via **SUSE:SLE-11-SP2:Update:Test** project. You can branch those packages from that project and submit your updates to that project. Submissions to :Update:Test are handled via maintenance and security teams, so keep them informed via NEEDINFO in bugzilla (maint-coord@suse.de for non-security, security-team@suse.de for security). SWAMPIDs are still assigned and patchinfos are needed. For security updates, usually the security team writes the patchinfos if the "VUL-0" bug is reassigned to security-team@suse.de in bugzilla. For non-security updates we appreciate a submission of a patchinfo via SWAMP, but can also help with handling patchinfos on an individual base.

Similarly, packages that have been checked into **SUSE:SLE-11-SP1:GA** are maintained via **SUSE:SLE-11-SP1:Update:Test** project. If there is no package with the same name in SUSE:SLE-11-SP2:GA or SUSE:SLE-11-SP2:Update:Test, any changes to SP1:Update:Test will be made available by the maintenance coordination for all affected SP1 based products as well as all SP2 based products. This is independent on whether they follow the NMM channel model or the traditional channel model. Exactly the same will happen in both cases.

For packages that have been "branched" with SP2:GA, the packager needs to submit separate sources for SUSE:SLE-11-SP2:Update:Test in order to provide the packages that will be applied to SP2 based products.

What to keep in mind when doing changes in SP1:Update:Test?

Similarly to the situation with SLE10, all changes in SP1 are ensured by maintenance to be provided also for SP2 customers. As long as there are no separate sources in SP2:Update:Test, the identical build result will be used for SP2 as well.

This means in particular for packages not branched in SP2:

- Packages are built in a SP1 environment
- They are not rebuild against SP2 environment for SP2 products
- The binaries from the SP1 built are QAed and tested on both SP1 and SP2
- If the packages are determined to work on both SP levels, the update is released

This means that if a package needs a particular change that is only available in the SP2 build environment, you need to clearly document the reason for this in the buildrequires via a version dependency and ideally also explain the reason for that in the .changes file if obvious or in the bugzilla entry related to this change in greater detail (feel free to use a private comment for that).

Please keep in mind that versioned dependencies in BuildRequires are not automatically transferred to Requires for runtime dependencies. You often need in such cases also think about how to express a correct runtime dependency to ensure that the package can only be installed in valid scenarios.·

In any case of doubt, please drop a mail to maint-coord@suse.de to reach out for help.

What can I do to do feature updates between Service Packs?

Feature updates can be done between service packs upon Product Management request via FATE. In general, those features need to be unintrusive for the customer and have a low risk of regressions. Feature updates are evaluated and should be discussed in detail, so that the risks and benefits are determined and accepted by the appropriate stakeholders (Product Management, Release Managers and Maintenance, mainly).

You can help here by providing the valuable knowledge in researching if a particular update is:·

- fully binary backward compatible
- fully behaviorally backward compatible

How to do all that with the build service ?

With the deployment of OBS 2.3 in the internal build service, the osc commands "osc sm" (for searchmaintained) and "osc mbranch" (for branch maintained packages) have become a lot more useful for SLE maintenance.

with osc 0.134 or newer the commands should give you valid information about what to do:

```
$ iosc sm aaa_base
...
```

```
SUSE:SLE-11-SP1:Update:Test/aaa_base
SUSE:SLE-11-SP2:Update:Test/aaa_base
...
```

This is an example of a package that has been branched for SP2:GA, which means that since that point in time two package versions exist that need to be maintained separately. After the end of parallel maintenance of SP1, only the SP2 variant of the package will be returned by "iosc sm" and consequently, you can stop updating the SP1 based variant.

```
$ iosc sm curl
...
SUSE:SLE-11-SP1:Update:Test/curl
...
```

This is an example of a package that has not been branched for SP2:GA, which means any successful submission (that is accepted, QA'ed and released by Maintenance) will be made available to all SP1 based products that ship this package as well as all SP2 based products at the same time.

You can use these commands to prepare a maintenance update:

```
$ iosc bco SUSE:SLE-11-SP1:Update:Test curl
...
$ osc vc
...
$ osc ci
...
$ osc sr
```

In case of packages that require frequent changes, we urge you to set a devel project for your package on that code stream (similar to how it works in openSUSE:Factory) via the changedevelrequest. this simplifies repeated submissions and allows you collaborate on changes prior handing it off for maintenance. In such cases we also recommend you to link or branch the origin package, so that you get proper %release numbers assigned by the build service for your test packages.