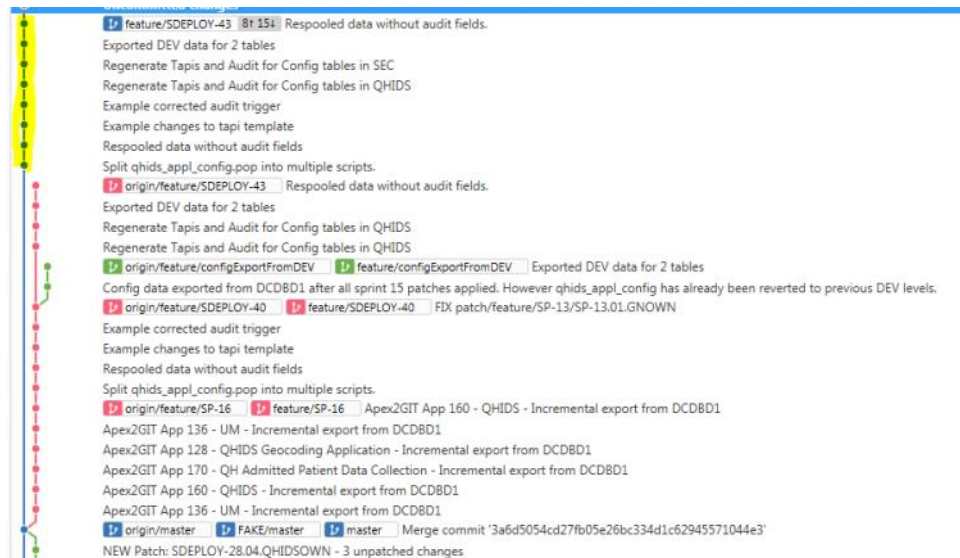# Tutorial - GitPatcher basics

Thursday, 21 March 2019   12:36 PM

Patching a completed feature

This feature has already been rebased and is ready for patching.
(Rebasing is still done automatically as a step during patching).

Review Tags

This is a verification step to allow you to be sure of what commits will be included
In this case we are going to patch commits from SDEPLOY-43.01A to SDEPLOY-43.01B
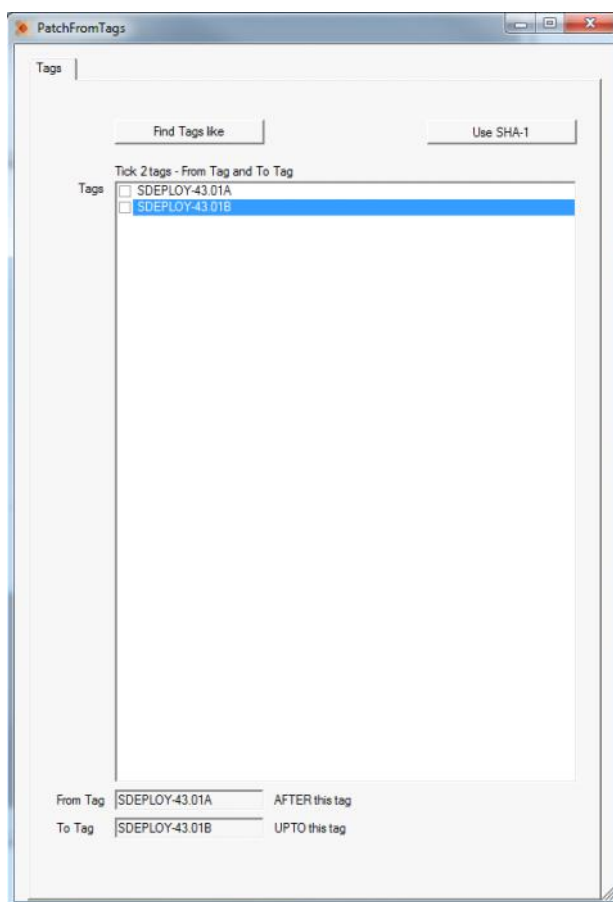Where those 2 commits are connected. 8 commits in all.
IE

- Excluding SDEPLOY-43.01A (Head of master)
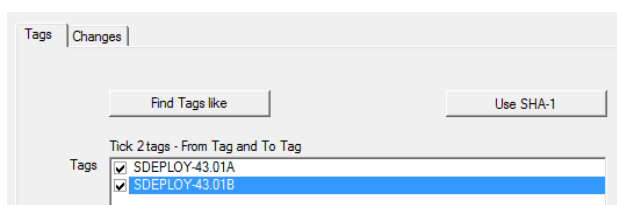- Including SDEPLOY-43.01B (Head of SDEPLOY-43)
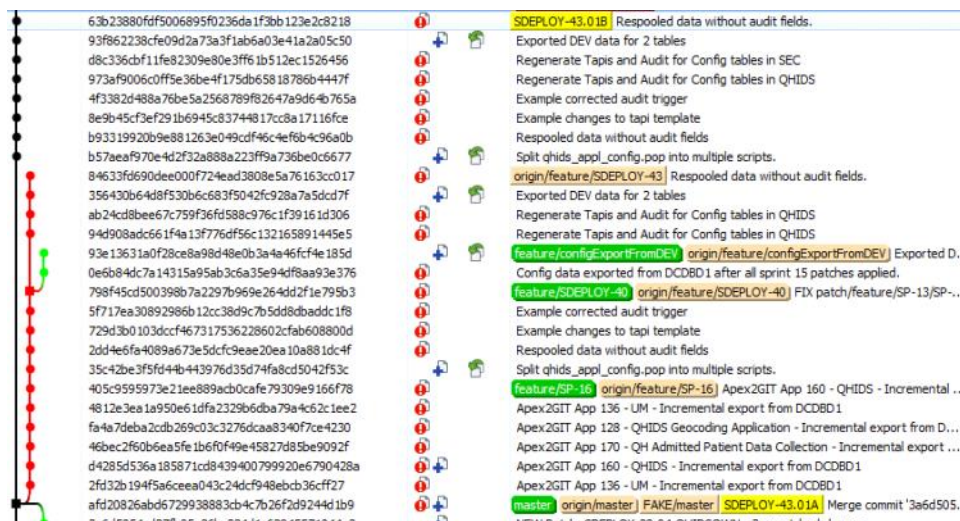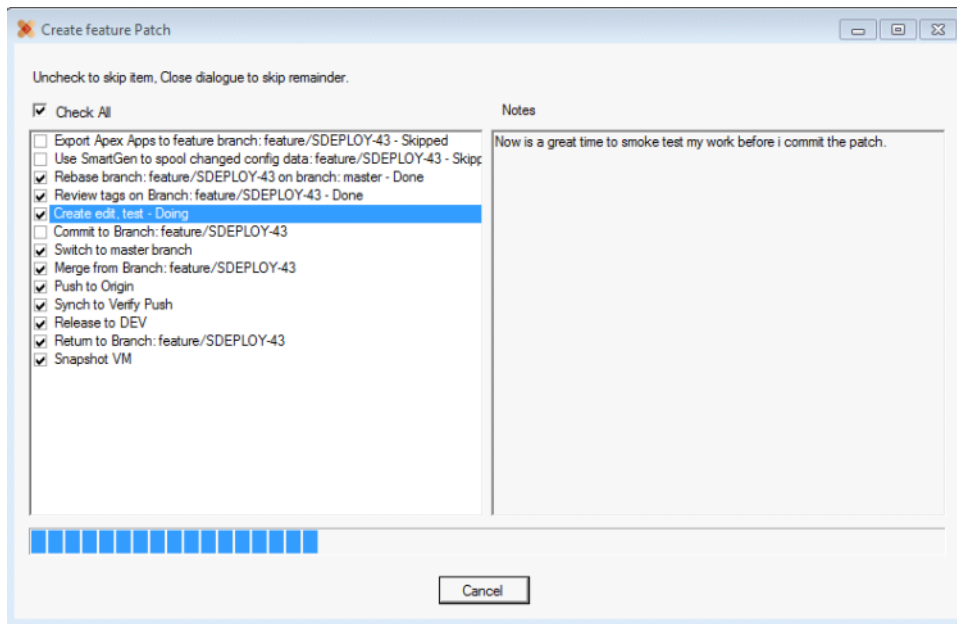
Opens the PatchFromTags dialog.

Tags tab

Pick the latest 2 tags A and B for the same patch no.

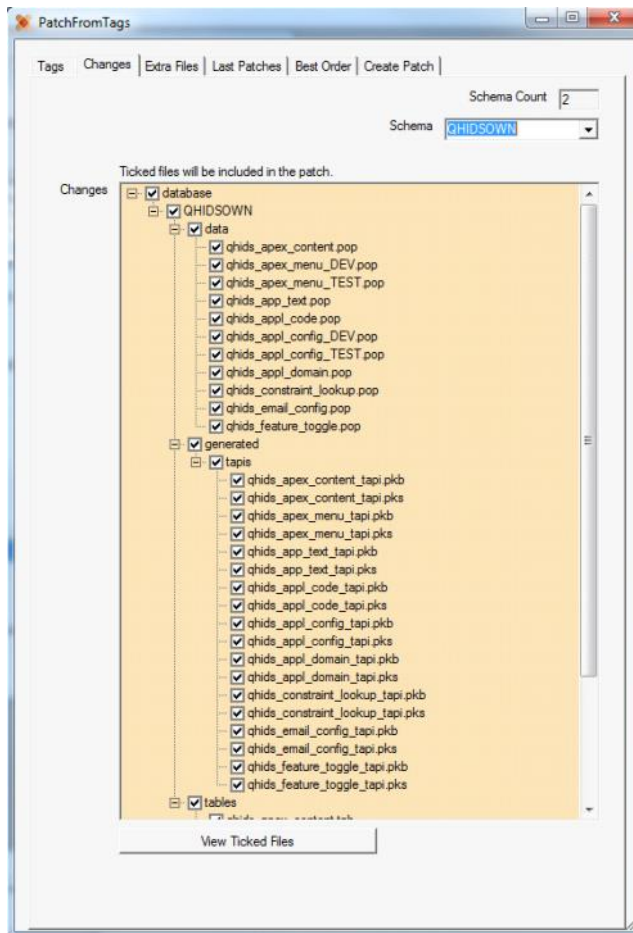| From Tag	SDEPLOY-43.01A	AFTER this tag
To Tag	SDEPLOY-43.01B	UPTO this tag

Changes tab is shown

Changes tab

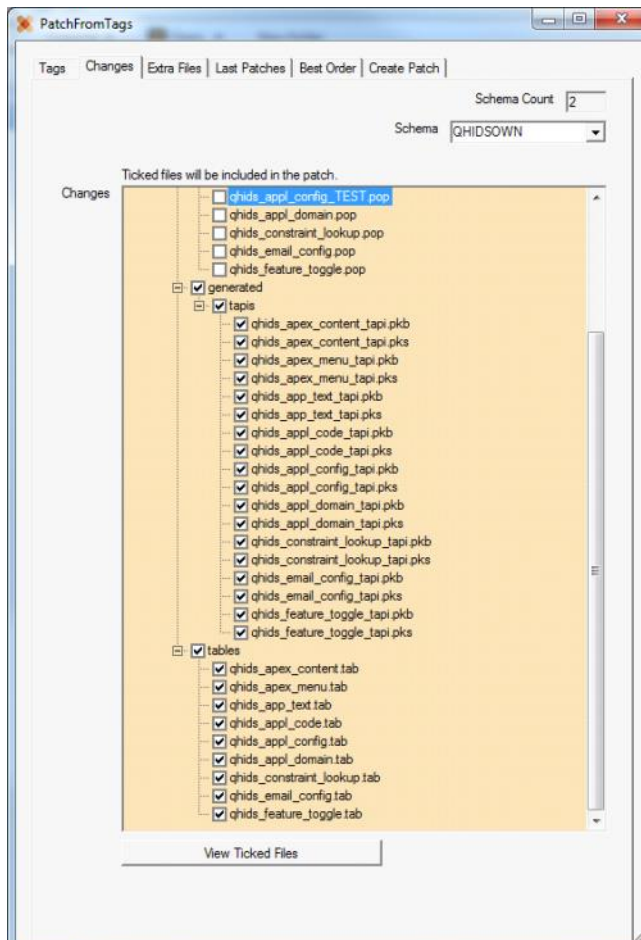Warning may appear - at least 2 patches will be needed.
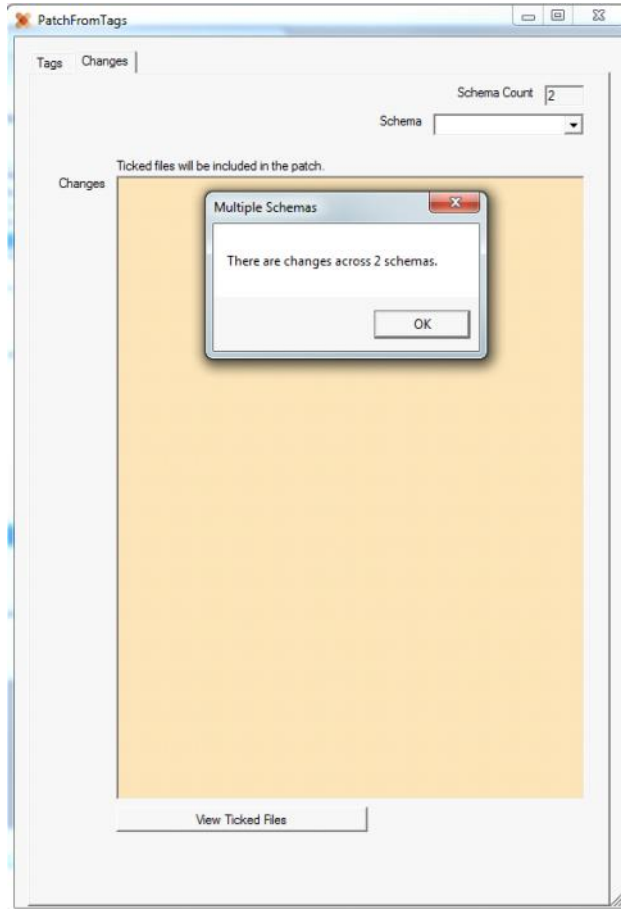
Choose a schema for first patch.

The remaining tabs are shown.
QHIDSOWN objects that were changed between the chosen tags

In this case I would like to patch the data in a separate patch, to give me a more flexible deployment. So I untick the data node.

Extra Files tab

Occassionally you may wish to include a file that was not changed.
Eg perhaps a view with select * must be rerun. This can be chosen here.
Take care to pick files that are for this schema (QHIDSOWN)
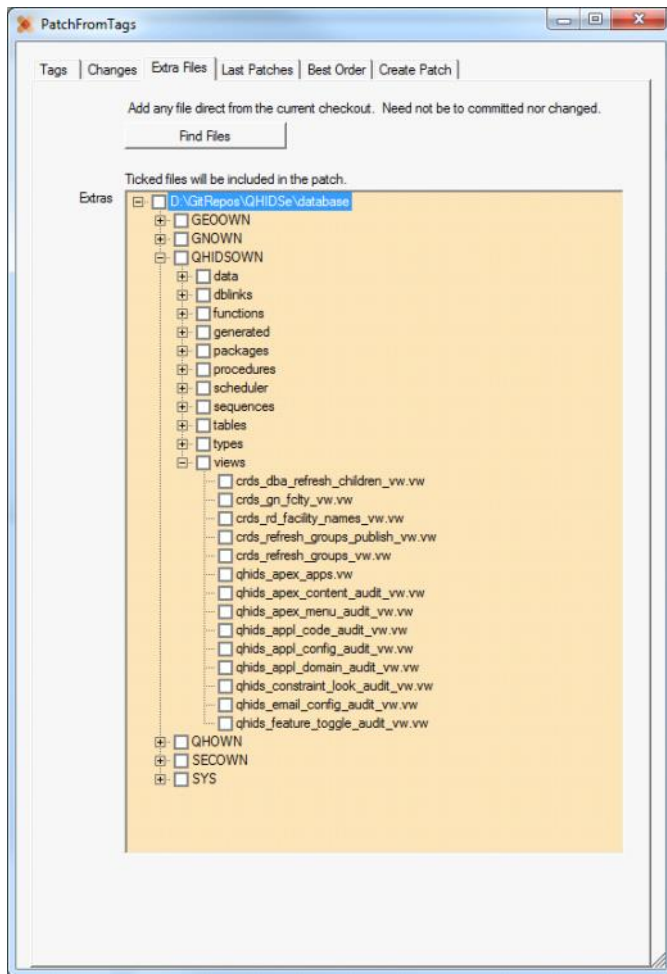
I won't choose any this time. But if I did (tick any) they would be included in the patch.

Last Patches tab
Please visit this page.
On opening the tab will select a set of patches as prerequisites of this patch.
For each component in this patch, it will find the last patch in which that component was included.
It interrogates the local database to discover this. This patch is then added to the set.
The set is displayed as ticked nodes in the tree of all patches found in the local git checkout of QHIDSe.
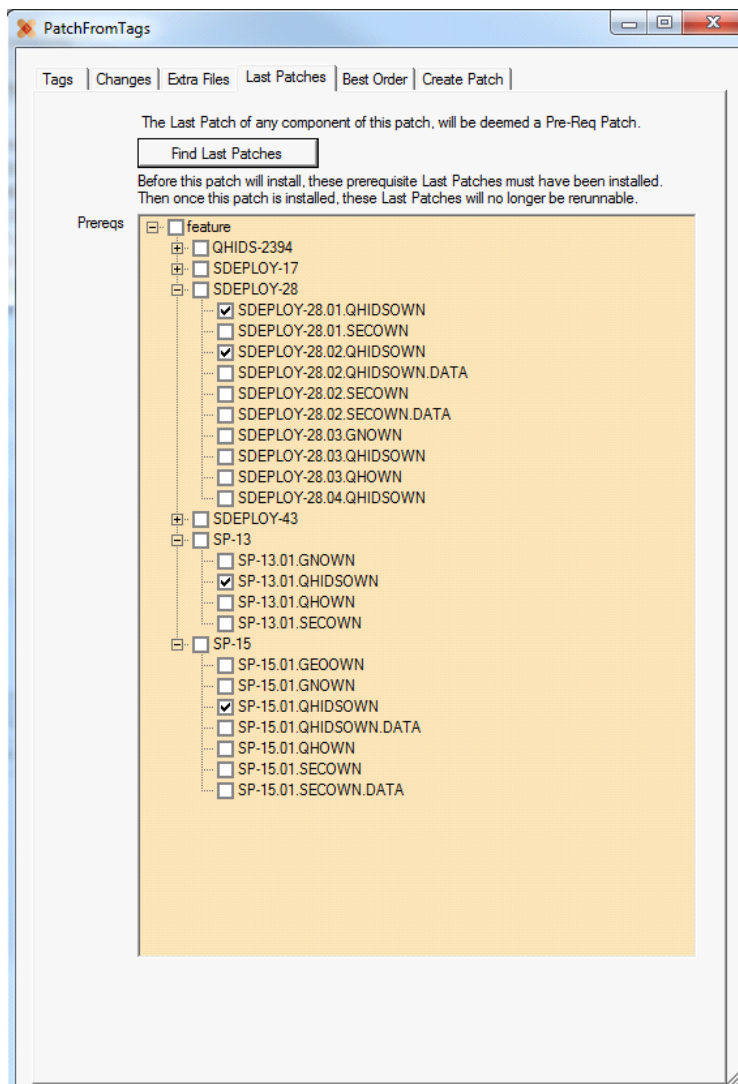
Best Order tab

This tab allows the developer to choose additional prerequisites that they know must be applied before the current patch.

Assuming the patches are built and deployed to the local DB, as you go in the best order, then you select at least the last patch you built here.
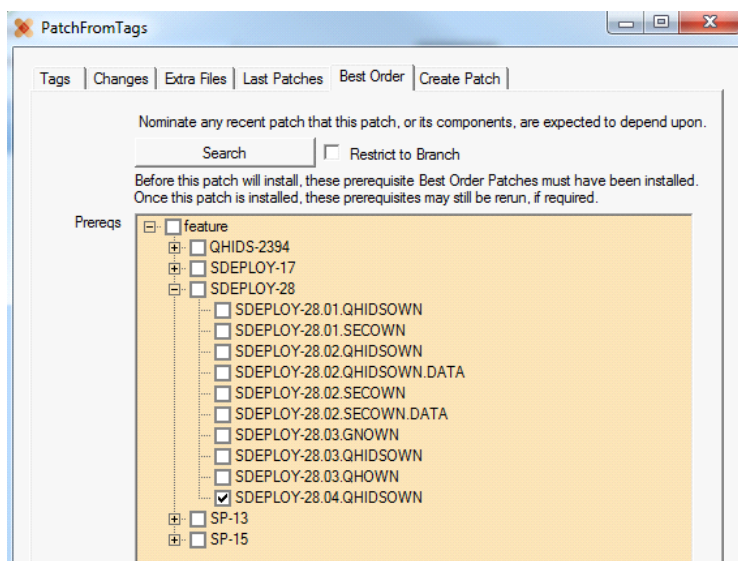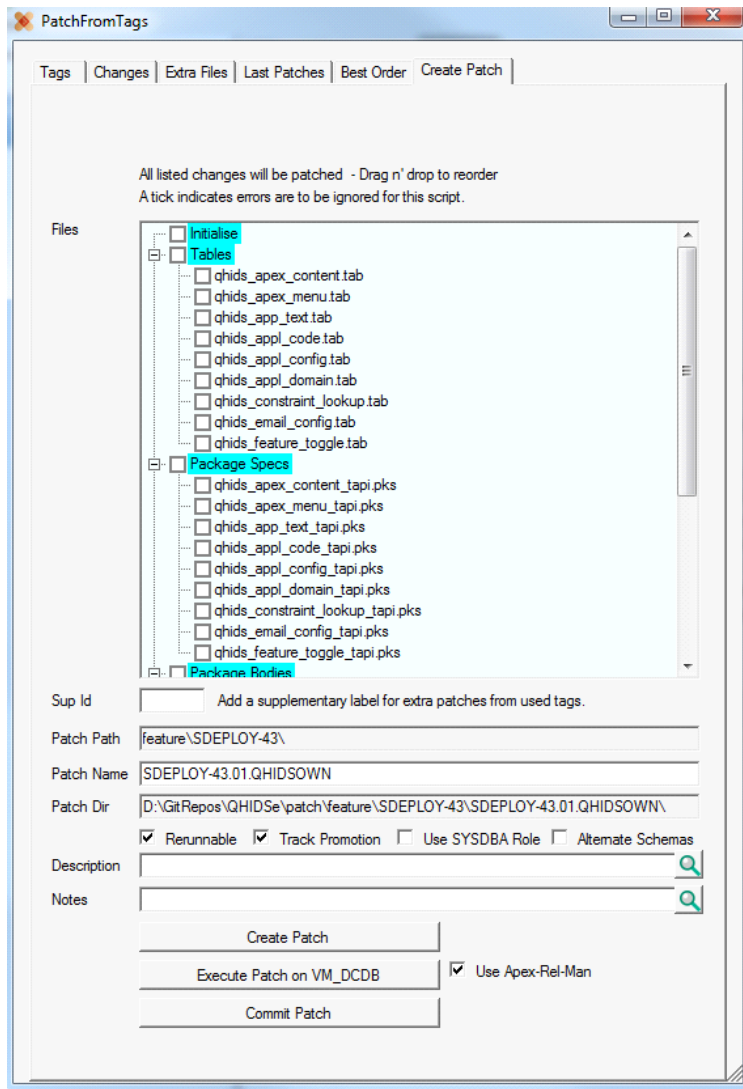

Create Patch tab

This is where is all comes together.

All the selected objects appear in another tree.
This tree allows for drag'n'drop reordering of files and sections.

Sections

The sections may be dragged into any order, but will always stay at the first level.
The only special section is the "Do Not Execute" node.
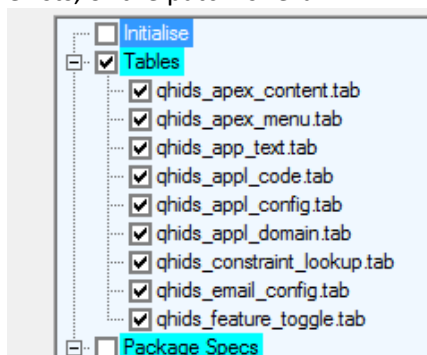Any file under this node will be included in the patch, but will not be installed.

Files

Files may be dragged into any order, in any section.
The tree always has 2 levels.

Tick files to Ignore Errors

Tick the Tables node, and any other files that you would expect to raise error if the object already exists, or the patch is rerun.

This allows the patch to be rerunnable.

Sup Id
- null this time

Patch Path
- Patch Path this is the relative path of the folder, in which the patch will be created.

Patch Name
- Patch Name - derived but editable.
- Must be unique among all patches to be installed in the database.
- For this reason it includes Feature name (Jira SDEPLOY-43), a feature patch sequence 01, schema name QHIDSOWN (and Sup Id if given).

Tick Boxes - ignore for now.
- By default a patch is flagged as Rerunnable (ONLY if the patch must NEVER be run twice, untick this box).
- By default a patch is flagged to Track its Promotion between databases
- Use SYSDBA Role for SYS, or if another user with similar privs is used for SYS operations.
- Alternate Schemas will allow the user to enter an alternative schema name, instead of a generic schema name.

Description - mandatory
Enter a patch description or Pick a heading from one of the included commits, with the drop down. Edit after selecting.

Notes - optional
Enter a patch noten or Pick a heading from one of the included commits, with the drop down. Edit after selecting.

Create Patch
The patch Folder is created and the files are exported into it.
- The patched files are as-at the second tag in the repo.
- Except for Extra Files, which are taken directly from the checkout.

ne	Date modified	Type	Size
install.sql	21/03/2019 1:59 PM	SQL File	7 KB
install_lite.sql	21/03/2019 1:59 PM	SQL File	4 KB
qhids_apex_content.tab	21/03/2019 1:59 PM	TAB File	3 KB
qhids_apex_content_tapi.pkb	21/03/2019 1:59 PM	PKB File	71 KB
qhids_apex_content_tapi.pks	21/03/2019 1:59 PM	PKS File	40 KB
qhids_apex_menu.tab	21/03/2019 1:59 PM	TAB File	10 KB
qhids_apex_menu_tapi.pkb	21/03/2019 1:59 PM	PKB File	170 KB
qhids_apex_menu_tapi.pks	21/03/2019 1:59 PM	PKS File	89 KB
qhids_app_text.tab	21/03/2019 1:59 PM	TAB File	5 KB
qhids_app_text_tapi.pkb	21/03/2019 1:59 PM	PKB File	92 KB
qhids_app_text_tapi.pks	21/03/2019 1:59 PM	PKS File	49 KB
qhids_appl_code.tab	21/03/2019 1:59 PM	TAB File	3 KB
qhids_appl_code_tapi.pkb	21/03/2019 1:59 PM	PKB File	69 KB
qhids_appl_code_tapi.pks	21/03/2019 1:59 PM	PKS File	39 KB
qhids_appl_config.tab	21/03/2019 1:59 PM	TAB File	4 KB
qhids_appl_config_tapi.pkb	21/03/2019 1:59 PM	PKB File	68 KB
qhids_appl_config_tapi.pks	21/03/2019 1:59 PM	PKS File	39 KB
qhids_appl_domain.tab	21/03/2019 1:59 PM	TAB File	2 KB
qhids_appl_domain_tapi.pkb	21/03/2019 1:59 PM	PKB File	42 KB
qhids_appl_domain_tapi.pks	21/03/2019 1:59 PM	PKS File	24 KB
qhids_constraint_lookup.tab	21/03/2019 1:59 PM	TAB File	3 KB
qhids_constraint_lookup_tapi.pkb	21/03/2019 1:59 PM	PKB File	54 KB
qhids_constraint_lookup_tapi.pks	21/03/2019 1:59 PM	PKS File	31 KB
qhids_email_config.tab	21/03/2019 1:59 PM	TAB File	3 KB
qhids_email_config_tapi.pkb	21/03/2019 1:59 PM	PKB File	55 KB
qhids_email_config_tapi.pks	21/03/2019 1:59 PM	PKS File	32 KB
qhids_feature_toggle.tab	21/03/2019 1:59 PM	TAB File	4 KB
qhids_feature_toggle_tapi.pkb	21/03/2019 1:59 PM	PKB File	67 KB
qhids_feature_toggle_tapi.pks	21/03/2019 1:59 PM	PKS File	38 KB

Execute on VM_DCDB

Every patch will ask for the schema password.
You also me asked for schema names for generic patches.

If you get the answer wrong. Just close the dialog, and relaunch it.

The patch will then run spooling output to a screen and a log file in the patches dir.

At the end of the patch, all invalid objects in the schema are recompiled,
and then any objects still invalid are listed.
Followed by a count or warnings and errors.
Below is the best result.

```
Remaining invalid objects in schema QHIDSOWN

PL/SQL procedure successfully completed.

Completed SDEPLOY-43.01.QHIDSOWN at 21-MAR-2019 04:08
Warnings :0
Errors   :0


PL/SQL procedure successfully completed.
```
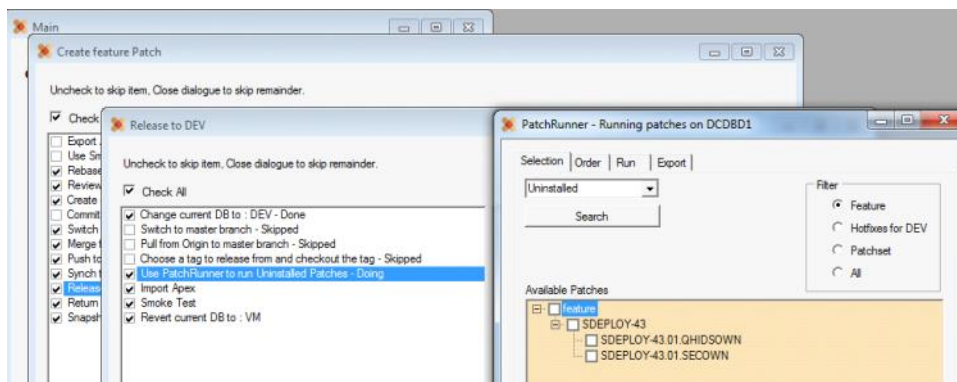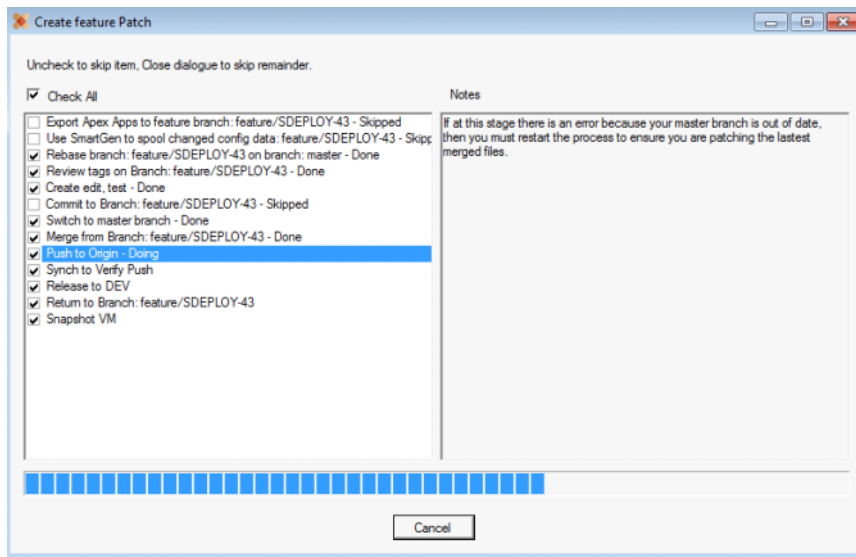
Close the window to continue patching.

If there are more patches to be created from the 2 chose tags, then return to the changes tab, and select the next set of changes.
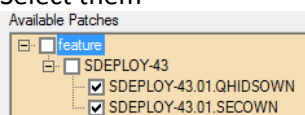
After the patches have been done, close the dialog, and the workflow will continue to merge the feature branch.
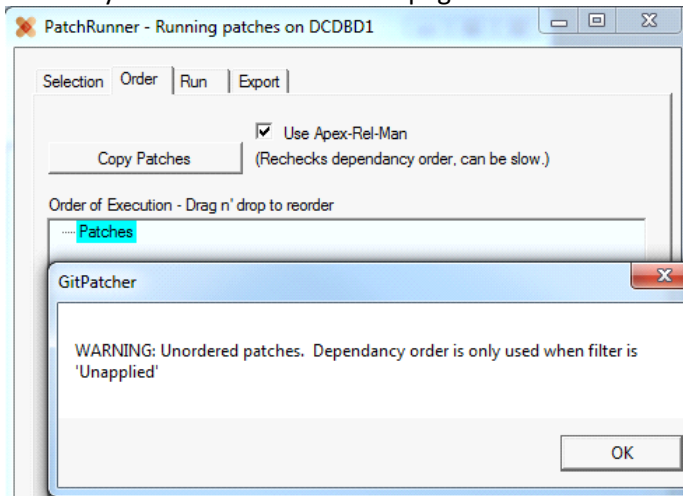

Patch Runner shows those patches uninstalled in DEV.

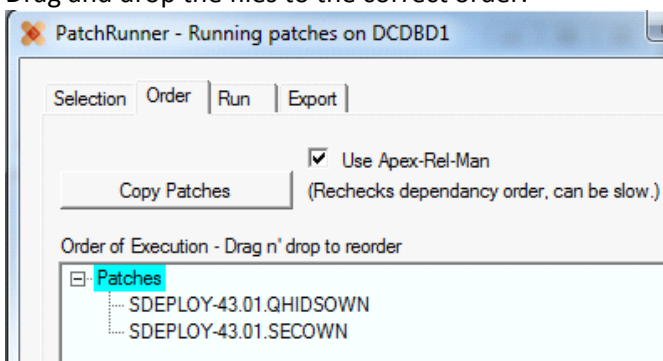*Selection tab*

Select them

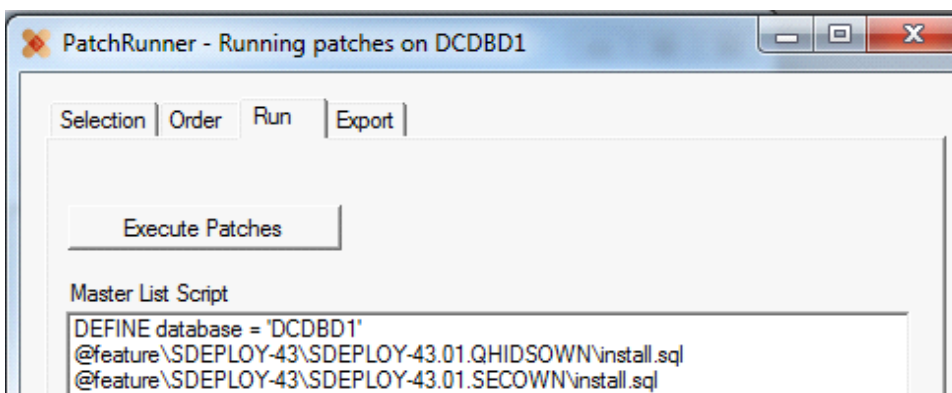*Order tab*

Manually order them on the next page.

DEV does not know of the existence of the VM, so it cannot find out the order. There is no DB link from DEV to any of the VM's.

Drag and drop the files to the correct order.

Run tab

This tab displays a master script that is used to install the patches.

Execute Patches

```
C:\windows\system32\cmd.exe - C:\oracle\sqlcl\bin\sql.exe  /nolog @D:\GitRepos\Q

SQLcl: Release 18.2 Production on Thu Mar 21 14:40:30 2019

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

LOG TO SDEPLOY-43.01.QHIDSOWN.log

SQL> SET FEEDBACK ON
SQL> SET PAUSE OFF
SQL> SET SERVEROUTPUT ON SIZE 1000000
SQL> SET TERMOUT ON
SQL> SET TRIMOUT ON
SQL> SET VERIFY ON
SQL> SET trims on pagesize 3000
SQL> SET auto off
SQL> SET verify off echo off define on
Enter value for QHIDSOWN_password: apple0wn
Connected.
Enter value for APEXRM_user: apexrm
```