

Creating a App Only Patch

Thursday, 28 March 2019 10:07 AM

Apex Apps and GitPatcher Patches

Apex Apps now "included" in GitPatcher Patches.

Actually the App is not INCLUDED, but it is QUEUED. This is to ensure each app is only run once, as part of a set of released patches.

GitPatcher patches can now be created with instructions to QUEUE an apex app to be installed, after the patches have run. There is a new tab in the PatchFromTags dialog to allow the developer to add changed Apps.

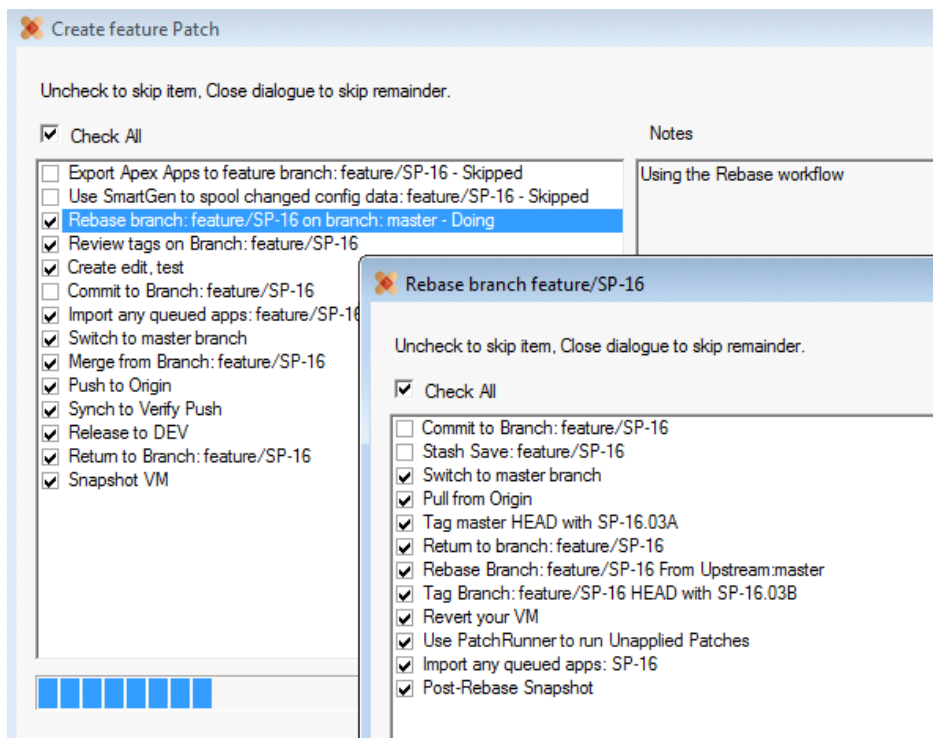
There is also a new dialog ApexAppInstaller, that interrogates the Queue and facilitates the Import of the Apex Apps.

NB: The database receiving your patches (be it a VM, or DEV, TEST etc) will now only be interrogating the queue to determine whether to install an app. So, if you change an App, this is now the way to ensure your changes are installed in other databases. Without this step other developers will not install your new version.

App-Only Patches

You can even create a Patch that ONLY queues Apex Apps and has NO database changes.

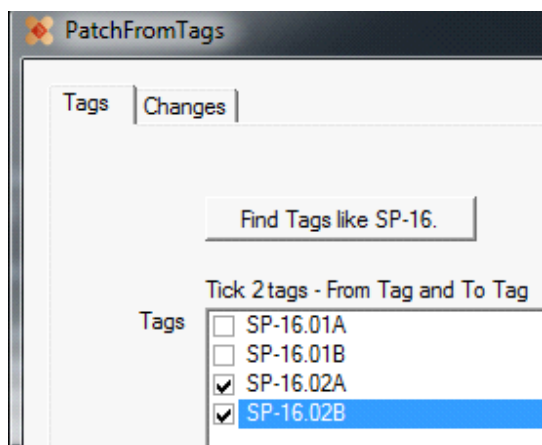
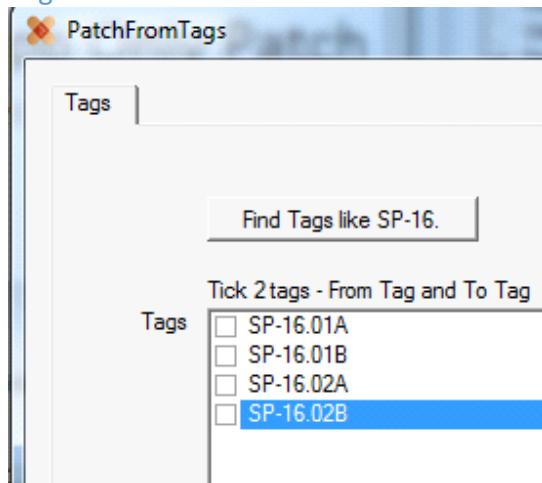
Please see the process below.



Follow the Rebase workflow to rebase the feature branch, as normal.

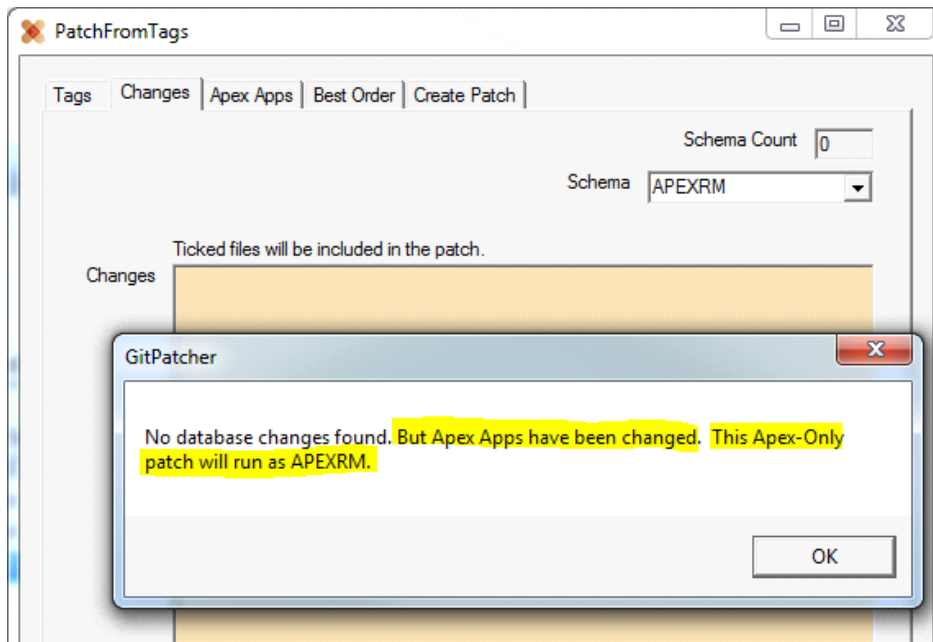
Create Patch From Tags

Tags tab



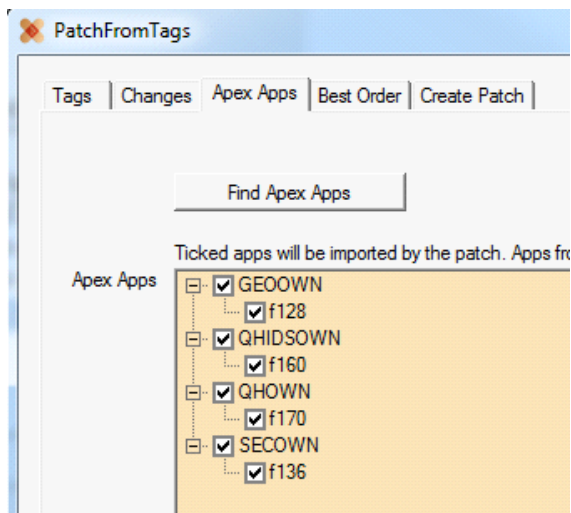
- Choose the tags

Changes tab



Acknowledge the warning. - OK
 There will be no files.
 Patch will run as APEXRM.

Apex Apps tab - THIS IS NEW.



All Apex Apps that were changed between the tags, will be listed, and ticked.
 Untick only if you do not want the record queued. (Usually you will just leave this as is)

You MUST visit the Apex Apps tab to populate the Apex Apps tree.

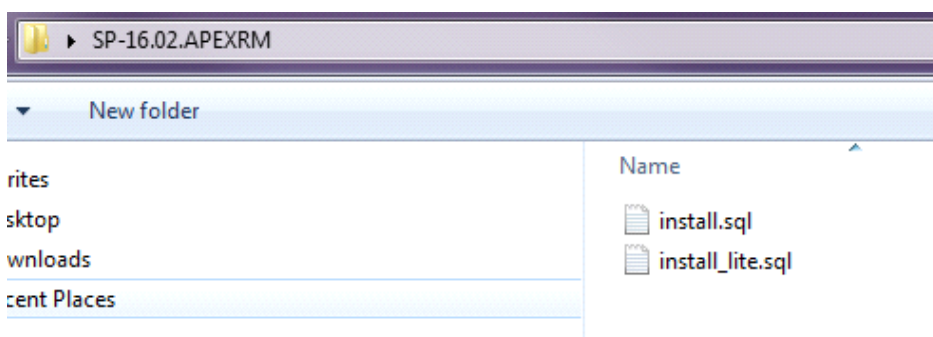
Extra Files tab is hidden - not relevant.
 Last Patches tab is hidden - not relevant.
 Best Order tab is still available, but optional

Create Patch tab

Description = Apex Apps Only

Notes = List of the apps included, to be queued.

Create Patch



- NB - Includes only the install files.

Execute the Patch

Answers

```
Enter value for APEXRM_password: apexrm
Connected.
Enter value for APEXRM_user: apexrm
```

Results

```
Remaining invalid objects in schema APEXRM

PL/SQL procedure successfully completed.
Enqueue Apex App 128
QUEUED APEX APP: 128

PL/SQL procedure successfully completed.
Enqueue Apex App 160
QUEUED APEX APP: 160

PL/SQL procedure successfully completed.
Enqueue Apex App 170
QUEUED APEX APP: 170

PL/SQL procedure successfully completed.
Enqueue Apex App 136
QUEUED APEX APP: 136

PL/SQL procedure successfully completed.
Completed SP-16.02.APEXRM at 28-MAR-2019 00:11
Warnings :0
Errors :0
```

Commit Patch

Files are already added, just hit commit.

Commit to: feature/SP-16

Message:

NEW Patch: SP-16.02.APEXRM - SP-16 Apex Apps Only

☐ Amend Last Commit

☐ Set author date

☐ Set author

Changes made (double-click on file for diff):

Check: **All** None Unversioned **Versioned** Added Deleted M

Path

☒ patch/feature/SP-16/SP-16.02.APEXRM/install.sql

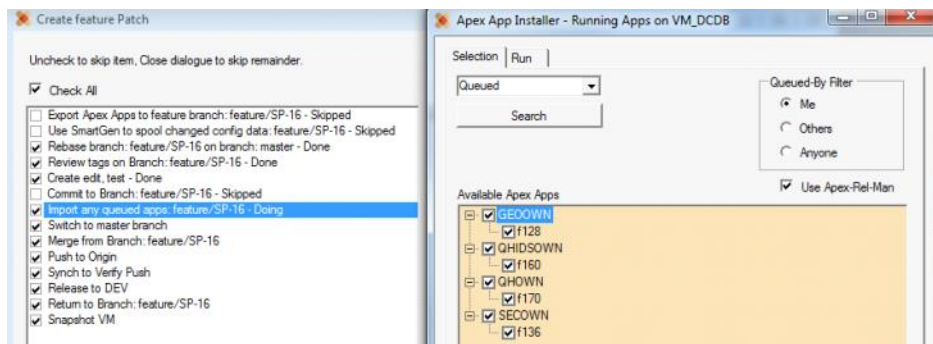
☒ patch/feature/SP-16/SP-16.02.APEXRM/install_lite.sql

Apex App Installer

This NEW dialog is included in 3 spots in the typical Feature Patch workflow. Always immediately after patches have been executed.

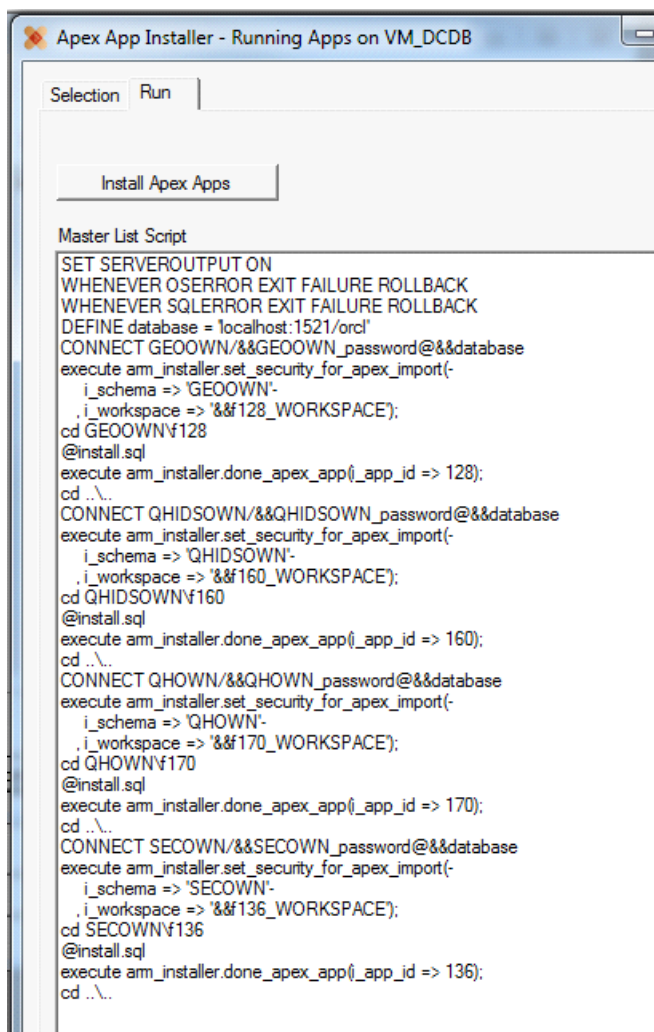
Selection tab

Shows all Apps queued, by the current OSUSER.



Run tab

This tab constructs a temp Master List Script.



- Install Apex Apps

The Master List Script is written to the filesystem and executed.

Answers

```

Enter value for GEOOWN_password: oracle
Connected.
Enter value for f128_WORKSPACE: QHIDS
PL/SQL procedure successfully completed.
API Last Extended:20180524
Your Current Version:20180524
This import is compatible with version: 20180524
COMPATIBLE <You should be able to run this import without issues.>

```

If you a non-existent workspace - will get NO DATA FOUND.

Results

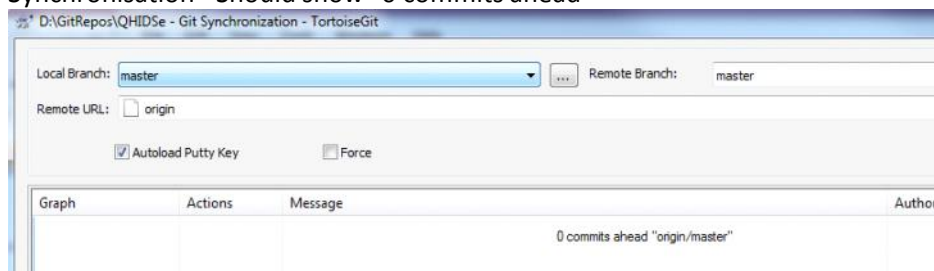
```
...done
DONE APEX APP: 128

PL/SQL procedure successfully completed.
Enter value for QHIDSOWN_password:
```

- ☐ Export Apex Apps to feature branch: feature/SP-16 - Skipped
- ☐ Use SmartGen to spool changed config data: feature/SP-16 - Skipped
- ☒ Rebase branch: feature/SP-16 on branch: master - Done
- ☒ Review tags on Branch: feature/SP-16 - Done
- ☒ Create edit, test - Done
- ☐ Commit to Branch: feature/SP-16 - Skipped
- ☒ Import any queued apps: feature/SP-16 - Done
- ☒ Switch to master branch - Done
- ☒ Merge from Branch: feature/SP-16 - Doing
- ☒ Push to Origin
- ☒ Synch to Verify Push
- ☒ Release to DEV
- ☒ Return to Branch: feature/SP-16
- ☒ Snapshot VM

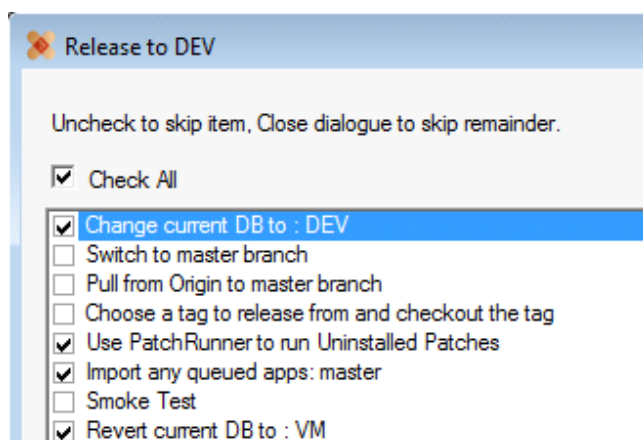
Merge and Push are now automatic, unless there are errors.

Synchronisation - Should show "0 commits ahead"



Release to Dev

As a general rule, all patches merged to master should also be released to DEV. Otherwise other developers will not apply the patches to their databases, and databases will get out of synch with the master branch.

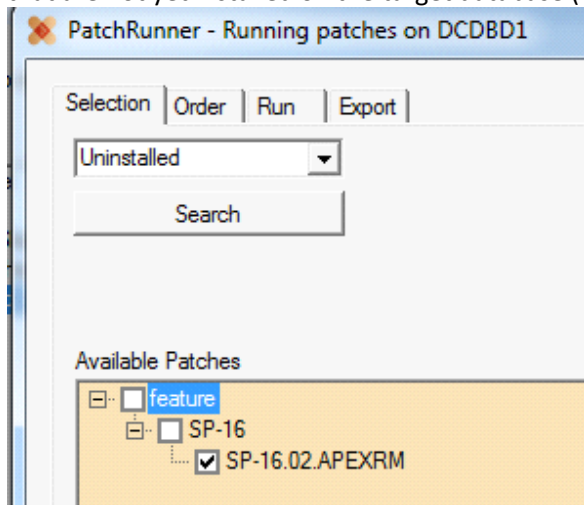


PatchRunner

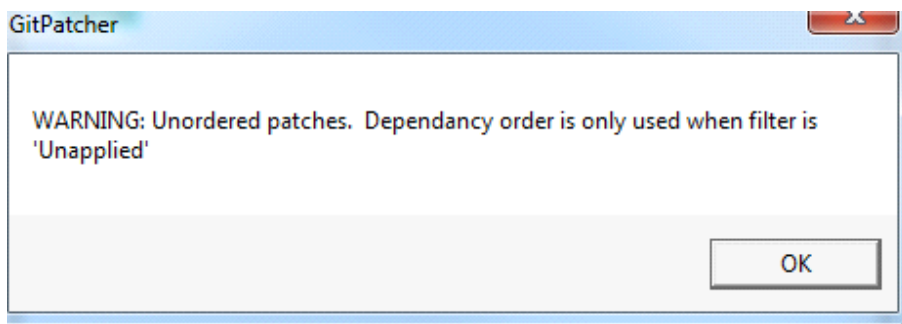
Selection

Install the "Uninstalled" patches to DEV.

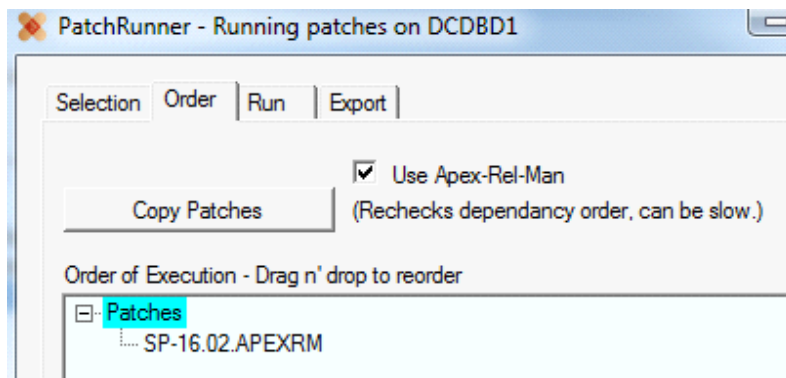
These are a list of all available patches (in the current GIT branch) that are not yet installed on the target database (DEV).



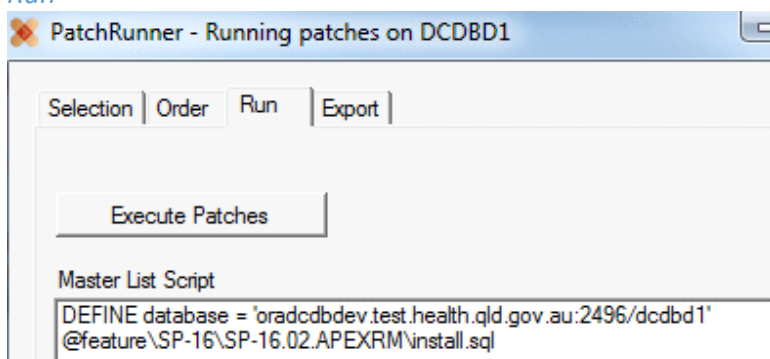
Order



If there is more than one, you must order these correctly yourself. Drag and Drop, to the preferred order.



Run




```

PL/SQL procedure successfully completed.

Enqueue Apex App 136
QUEUED APEX APP: 136

PL/SQL procedure successfully completed.

Completed SP-16.02.APEXRM at 28-MAR-2019 10:20
Warnings :0
Errors :0

PL/SQL procedure successfully completed.

Commit complete.

install.sql - COMPLETED.

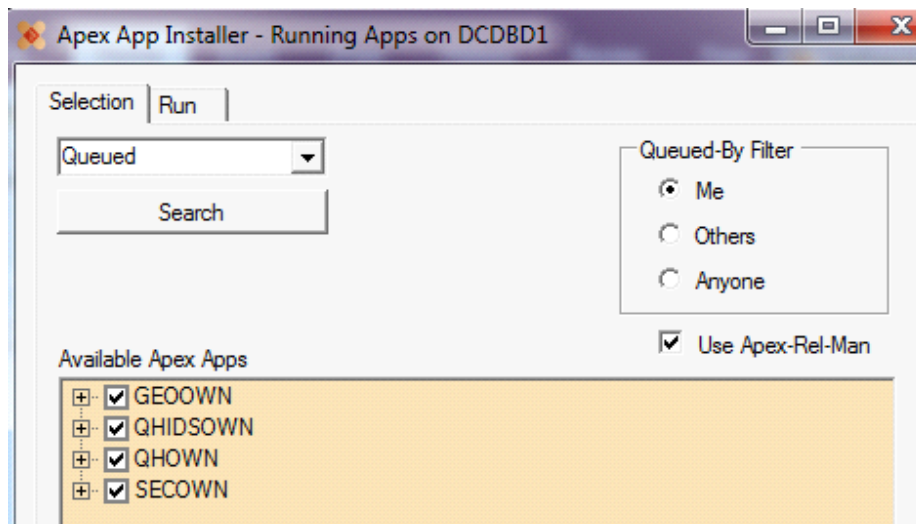
Commit complete.

SQL>

```

Import Queued Apps

Now user Apex App Installer to promote the Apps to DEV



Snapshot

Take the final snapshot.

Remove any leaf snapshots that are no longer needed.

