

pdsynch

Utility to :

1. Synchronize a list of z/OS Datasets (PDS) from one Master LPAR to one or many others.
 - Dataset level : Created Datasets under the HLQs in the list will be created in the remote LPARs. All members will be populated as well. Deleting a Dataset in the Master LPAR **will not delete** Datasets in the remote LPARs.
 - Member level : Member creation, deletion or updates will be reflected in the remote LPARs.
2. Synchronize a list of z/OS Datasets from the Master LPAR with a GitHub repo automatically. Every synchronization cycle will create a commit version of each member (v#m#).
3. Git clone the GitHub repository in a local workspace, create, delete or update files and synchronize the changes with all LPARs after committing the changes back into GitHub.
4. Use one way synchronization LPAR -> GitHub or GitHub -> LPAR or bi-directional LPARs - GitHub

At this particular case we work in a master LPAR creating, deleting or updating members. The change will be synchronized in all remote LPARs defined in the [config.json](#) configuration file.

Requirements:

- Windows
 - zowe
 - zOSMF REST APIs installed on the Mainframe
 - ooRexx
 - local git global configured by any valid method:
 - git config --global user.name myusername
 - git config --global user.email myemail
 - git config --global github.user myusername
 - git config --global github.token mytoken
-

Instructions

- Plant a tree.
- Create an empty GitHub repo (only the first time).
- Configure your settings at [config.json](#) :
 - **hlq.#** : Set of libraries to synchronize. # must be sequential number (hlq.1, hlq.2 ...).
 - "hlq.1" : "CUST001.M*",
 - "hlq.2" : "CUST002.M"
 - **cycle** : Time in seconds between synchronization cycles. A value of 0 means a single execution.
 - "cycle" : "0" : Single cycle execution.

- `"cycle" : "60"` : Execute synchronization every minute.
 - `master_prof` : Zowe zosmf profile for the Master LPAR. This LPAR is where we will be updating PDSs.
 - `"master_prof" : "zosmf-sr01brs"`
 - `prof.#` : Zowe zosmf profiles for the remote LPARs where the synchronization is going to take place. # must be sequential number (prof.1, prof.2 ...).
 - `"prof.1" : "zosmf-pe01",`
 - `"prof.2" : "zosmf-pe02"`
 - `stop` : Set it to 'N' to run the utility. Change to 'Y' whenever you want to stop the synchronization correctly. It will stop the process in the next loop. This is to avoid shutting down the utility in the middle of the synchronization process.
 - `ghrepo` : GitHub repository to manage PDS members versioning.
 - `"ghrepo" : "https://github.com/drb1972/demo.git"`
 - `pds2git` : Enables Mainframe to GitHub synchronization. Whenever we create, update or delete a member in the Master LPAR the change will be committed in the GitHub repo.
 - `"pds2git" : "Y"`
 - `"pds2git" : "N"`
 - `git2pds` : Enables GitHub to Mainframe synchronization.
 - `"git2pds" : "Y"`
 - `"git2pds" : "N"`
- Execute st.rex

More info

- **pdsync** has been developed in ooRexx on purpose to ease the path of experienced mainframers towards modernization with open source tools.
- **pdsync** is set to synchronize only PDS or PDS-E files with LRECL 80, but can be set to synch any dataset on the mainframe. It is meant to version :
 - Home made ISPF applications that are not included in standard Mainframe SCMs (JCL, Rexx, parm, panels, messages, skeletons, etc).
 - Parameter Datasets for System libraries, ISV products, etc.
 - Procedure libraries shared accross the LPARs
 - Personal sysadmin libraries, utility libraries, cmdprocs, ISPF settings, etc.
- The time to complete one cycle of the process (check changes on Master LPAR datasets and populate them) will depend on the amount of datasets candidates for synchronization and the amount of target LPARs.
- All code is in <https://github.com/drb1972/pdsynch.git> . The repo is public
- When executing the `st.rex` script, the utility will create a service workspace directory named after the GitHub repo at C:\Temp.
- Synchronization is be done at member level, not the whole libraries.
- Very usefull when there is no shared DASD

Important Notes

- The first time executing the **pdsync** utility the candidate datasets for synchronization defined in the [config.json](#) file should only exist in the Master LPAR.
 - The first cycle of the first execution will take longer than the rest since **pdsync** needs to create all datasets and populate them into the target LPARs.
 - When working offline with git2pds synchronization, it is recommended to delete the cloned workspace after finishing and clone it again the next time or issue a "git pull" command before pushing the changes back into the GitHub repo to be synchronized with the LPARs.
 - When defining wildcards at "hlq.1" : "CUST001.M*", at the [config.json](#) only asterisks (*) are allowed at the last position of the dataset name.
-

Tests

Test 1 :

- Edit a member in 3270, delete a couple of lines and add another.
- Delete another member
- Create a new member
- Wait a minute before you go to GitHub or the other LPARs and see the changes. Edit the member in GitHub that you modified and click on history icon or in the commit code to track the changes.
- Repeat various times by changing the same member. You'll see the modification history.

Test 2 :

- Open VSCode (or your preferred IDE)
- Open your projects folder and open a terminal
- "git clone [your_github_repo](#)"
- Position your prompt in the cloned folder
- Modify, delete and create new files
- Update the GitHub repo with your preferred method (GH Desktop, VSCode extension, command line, ...)
 - git add .
 - git commit -m "[any-message](#)"
 - git push
- With this method, you will have one commit for all changes you made, so this is better if you want to group changes in a single commit
- Check the modifications in the LPARs

Test 3 :

- Create a PDS with the same HLQ than one that is being synch and feed it with some members
- Check it exists in GitHub and the target LPARs after waiting a few seconds to complete the cycle

Test 4 :

- Delete the PDS you just created
- Check it has disspeared from GitHub after waiting some seconds to complete the cycle
- This version do NOT delete the PDS from the target LPARs

Test 5 :

- Add a new hlq in config.json file
-

Any comment is welcome: diego.rodriguez@broadcom.com , and don't forget to plant the tree