Example using Git ISPF client and IBM DBB

Created by rbarosa@us.ibm.com – Reviewed by Liam Doherty July 22, 2020

Introduction

This paper is created to help starting using Git/ISPF, does not intend to replace any IBM documentation, but is as "as is" example.

In this paper I will provide an example using the <u>Mortgage application</u> provided by IBM as sample and using the z/OS running on my <u>ZDT</u> environment.

Git ISPF client provides an ISPF interface that interacts with a Git repository to allow cloning, staging, checking in, pushing and pulling as well as other git commands. Look at the Git/ISPF install and documentation at: https://github.com/IBM/dbb/tree/master/IDE/GitISPFClient

On our example using the Git ISPF the main tasks are:

- 1. Cloned the GitHub provided Mortgage COBOL application to the z/OS (Rocket Git)
- 2. Edited a COBOL program using the ISPF editor and changed a comment line.
- 3. Requested a DBB USER Build to compile and link the modified program using the provided Groovy Scripts by zAppBuild
- 4. Issued a Git Commit and Push the changes to the GitHub

Running the ISPF Git Client

Start the ISPF Git client dialog using the BGZGIT REXX executable code.

On my example:

(EX 'IBMUSER.GITISPF.SBGZEXEC(BGZGIT)')



Once you have started the ISPF Git client you should go to Option **0** - **Preferences**.

```
Menu Help

Git Interface Primary Option Menu
Option ===> 0

O Preferences Terminal and user preferences
1 Git Init Create a Git Repository
2 Repository Work with cloned Git repository
X Exit Exit client
```

I made no changes.. **PF3** to return.



1. Cloning a remote repository

To start working on code in a remote Git repository

1.1. Go to option 2 - Repository.

You will need to enter a repository address in the line at the top of the screen.

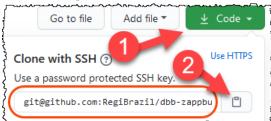


In my example I have already a repository cloned:



1.2. You may "Fork" the provided sample from https://github.com/IBM/dbb-zappbuild. Doing this, I created my repo at https://github.com/RegiBrazil/dbb-zappbuild-1 (this is Private repository that nobody has access).

Click on the green button and copy the repo location (must be SSH)



1.3. Since the repository name will be longer than the provided space, **put the cursor in the field, press PF4,** paste the repo name and press **PF3**:

```
BGZNREPO+0

Line 1 of 14

Scroll ===> PAGE

git@github.com:RegiBrazil/dbb-zappbuild-1.git
```

1.4. Paste the USS working directory (where the clone will live). I used /var/jenkins/gitispf.

Notice that you also may need to press PF4 if name is longer.

The directory you specify will have an extra directory level added of the repository being cloned. So in the example above, I entered /var/jenkins/gitispf and the working directory will be /var/jenkins/gitispf/dbb-zappbuild-1

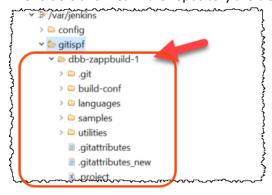
1.5 Once you press enter the repository will be cloned into the working tree directory.

```
C Cloning into 'dbb-zappbuild-1'...

E Cloning completed

f
```

The folders and files in the repository are cloned into the zFS and not into PDS/PDSE.



1.6. Press **PF3** and type / at the repository cloned. You may issue Git commands. For example check the status...



1.7. Type 2 to see the Git status of this repo:

1.8 Result will be:

```
Command ===> _____ Scroll ===> PAGE

On branch sandbox
Your branch is up-to-date with 'origin/sandbox'.
nothing to commit, working directory clean
```

2. Working with files in a remote repository

Once a repository is cloned you can choose to work on the files either through the ISPF Git interface or you can work on them directly through ISPF UDLIST (3.17), or using your editor of choice through OMVS...

There are many commands that you may use. See the online doc or use **PF1** for help.

2.1. Move the mouse to the field on left of the cloned repo and press **PF1** for the help.

You may use **PF8 and PF7** to advance and return.



2.2. To start working with files enter **JU** (Jump to Working Directory) next to the repository that contains your files.

2.3. To drill down into a directory enter an L next to the directory

```
git@github.com:RegiBrazil/dbb-zappbuild-1.git
Git Repository :
On Branch ..... :
                     sandbox
Pathname /var/jenkins/gitispf/dbb-zappbuild-1
     Filename
                                     Git Status
     build.groovy
                                                  File
     build-conf
                                                  Dir
     build2.groovy
                                                  File
     languages
                                                  Dir
    samples
                                                  Dir
     utilities
                                                  Dir
     BUILD.md
                                                  File
     CONTRIBUTIONS.md
                                                  File
     DC01.1.txt
                                                  File
     INSTALL.md
                                                  File
     LICENSE
     README.md
                                                  File
  *GIT
```

2.4. To drill down into a directory enter an L next to the directory

and

```
Git Repository
                     git@github.com:RegiBrazil/dbb-zappbuild-1.git
On Branch ....:
                     sandbox
Pathname /var/jenkins/gitispf/dbb-zappbuild-1/samples/MortgageAppl
     Filename
                                     Git Status
                                                  Type
     application-conf
                                                  Dir
     bms
                                                  Dir
     cobol
                                                  Dir
     copybook
                                                  Dir
     link
                                                  Dir
     Jenkinsfile
                                                  File
     README.md
```

2.5 To see the available options for the file level press **PF1**:

```
Help for Working Directory

More: +
This panel allows you to view, edit, and manage the file that match the specified pattern. Various information about the UNIX file is displayed in several columns:

P Type: The type of the UNIX file. Can be Dir (for a directory) or File.

Git Status: The Git Status of files which have been modified (changes not staged for commit), or staged (changes to be committed), in your local working directory.

Enter a '/' in the area next to a file or directory to see a menu of actions, or enter an action shortcut character from the following list:

Character Description

E Edit this file
```

2.6 To edit the COBOL using EBCDIC type E.

Note: probably would be easier here directly through ISPF UDLIST (3.17), or using your editor of choice through OMVS..

To create a new file enter S on the command line. Make sure you give the filename an extension that matches an entry in your .gitattributes file. This way, when the file is added to staging it will automatically be tagged with the correct CCSID.

```
Git Repository
                git@github.com:RegiBrazil/dbb-zappbuild-1.git
On Branch ....:
                sandbox
Pathname /var/jenkins/gitispf/dbb-zappbuild-1/samples/MortgageApplica
   Filename
                             Git Status
                                      Type
    epscmort.cbl
                                      File
    epscsmrd.cbl
                                      File
    epsmlist.cbl
    epsmpmt.cbl
                                      File
    epsnbrvl.cbl
```

Press enter

```
Command ===>

Object Name:
//var/jenkins/gitispf/dbb-zappbuild-1/samples/MortgageApp
Initial Macro . _____
Profile Name . . . _____
Profile Name . . . _____
Panel Name . . . _____
Panel Name . . . _____
Options _____
Confirm Cancel/Move/Replace _____ 1. ASCII
___ Mixed Mode _____ 2. UTF-8
__ Preserve VB record length

Press ENTER to continue. Press CANCEL to cancel action.
```

2.7. Make a small change in the code as below and press PF3

```
File Edit Edit_Settings Menu Utilities Compilers Test
                                                          <u>H</u>elp
          /var/jenkins/gitispf/dbb-zappbuild-1/samples/Mo Columns 00001
EDIT
-Warning- The UNDO command is not available until you change
               your edit profile using the command RECOVERY ON.
             ID DIVISION.
000002
            PROGRAM-ID. EPSCMORT.
              THIS DEMONSTRATES CICS/DEBUG
                                                    - EPSDEMOS 2008
           * changed July 21, 2020 by Regi

* THIS PROGRAM WILL RECEIVE A DATE AND COVERT THE DATE TO
000004
000005
                AN INTEGER IN A CALLED PROGRAM TO DETERMINE DAYS FROM
000007
                CURRENT DATE.
```

2.8. Notice that the status is changed to "Modified"

```
Git Repository
                     git@github.com:RegiBrazil/dbb-zappbuild-1.git
                     sandbox
Pathname /var/jenkins/gitispf/dbb-zappbuild-1/samples/MortgageApplication/
                                      Git Status
     Filename
                                                  Type
                                      Modified
     epscmort.cbl
                                                  File
     epscsmrd.cbl
                                                  File
                                                  File
     epsmlist.cbl
     epsmpmt.cbl
                                                  File
     epsnbrvl.cbl
                                                  File
```

2.9 Type **ST** to see the Git status.

```
Pathname /var/jenkins/gitispf/dbb-zappbuild-1/samples/MortgageApplication/

Filename Git Status Type

st epscmort.cbl Modified File

epscsmrd.cbl File

epsmlist.cbl File

epsmpmt.cbl File

epsnbrvl.cbl File
```

2.10 The result should be as below.

At this point you can try a DBB user build.. You do not need to commit the changes to Git server untilyou are happy with the results.

```
Command ===> _______ Scroll ===> PAGE

On branch sandbox
Your branch is up-to-date with 'origin/sandbox'.
Changes not staged for commit:
    (use "git add <file>..." to update what will be committed)
    (use "git checkout -- <file>..." to discard changes in working directory)

S modified: samples/MortgageApplication/cobol/epscmort.cbl
no changes added to commit (use "git add" and/or "git commit -a")
```

2.11 Press PF3 to dismiss the message

3. Request a DBB User Build through ISPF Git interface

You must enter **UB** (User Build) next to the file you want to build. On the DBB User Build panel, you will have to specify the build script to use (an existing DBB groovy build script that uses *IBM Dependency Based Build*), and the build destination HLQ where you want to have your file built.

3.1 Type **UB** for the COBOL program to be built using *User Build*.

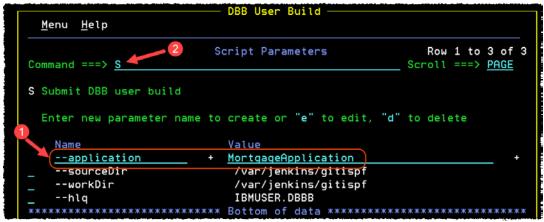
```
Git Repository : qit@qithub.com:ReqiBrazil/dbb-zappbuild-1.qit
On Branch ..... :
                sandbox
Pathname /var/jenkins/gitispf/dbb-zappbuild-1/samples/MortgageApplication/
                              Git Status
    Filename
                                       Type
<u>ub</u> ) epscmort.cbl
                              Modified
                                       File
    epscsmrd.cbl
    epsmlist.cbl
    epsmpmt.cbl
    epsnbrvl.cbl
                                       File
```

3.2 Specify the required data for the User Build /var/jenkins/gitispf/dbb-zappbuild-1/build.groovy /var/jenkins/gitispf /var/jenkins/gitispf/work IBMUSER.DBB

And press Enter to add more parameters..

```
DBB User Build
Command ===>
Build file epscmort.cbl
                         /var/jenkins/gitispf/dbb-zappbuild-1/build.gro
Build script to use .
Build sandbox folder . /var/jenkins/gitispf
Log file location . . /var/jenkins/gitispf
Build destination HLQ
                       IBMUSER.DBB
 (Fully qualified, no quotes)
View build output on completion /
Instructions:
 Enter S Command to request DBB user build.
  Press ENTER to set additional parameters to be used in script.
  Press CANCEL or EXII to cancel DBB user build.
 *GIT
```

3.3 Type -application parameter and S to submit



3.4 The *User Build* will start, and the messages will be displayed.

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
VIEW
          /var/jenkins/gitispf/dbbub.log
                                                         Columns 00001 00072
Command ===>
                                                            Scroll ===> PAGE
    -CAUTION- Data contains invalid (non-display) characters. Use command
                ===> FIND P'.' to position cursor to these
      -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
000001 JVMJ9VM090I Slow response to network query (61 secs), check your IP DNS
000002
000003
       ** Build start at 20200721.102817.028
000004 ** Build output located at /var/jenkins/gitispf
000005 ** Adding dbb-zappbuild-1/samples/MortgageApplication/cobol/epscmort.cb
000006 ** Writing build list file to /var/jenkins/gitispf/buildList.txt
000007
       ** Invoking build scripts according to build order: BMS.groovy,Cobol.gr
000008 ** Building files mapped to Cobol.groovy script
000009 *** Building file dbb-zappbuild-1/samples/MortgageApplication/cobol/eps
000010 ** Writing build report data to /var/jenkins/gitispf/BuildReport.json
000011 ** Writing build report to /var/jenkins/gitispf/BuildReport.html
000012 ** Build ended at Tue Jul 21 22:29:38 GMT 2020
000013 ** Build State : CLEAN
000014
       ** Total files processed : 1
```

Note: It may have strange non EBCDIC character on the log created. You can ignore that. But if you issue a **find P'.'** command you will see that there is a bad character at line 1 (CHARS X'04').

3.5. You may see the log and buildlist.txt at the provided uss directory

```
Remote Systems 3 461
■ EPSCMORT.log
                                                                            1JVMJ9VM090I Slow response to network quer
                                                                                                                                                        > 👺 /var/dbb
      1@PP 5655-EC6 IBM Enterprise COBOL for z A
                                                                                                                                                        > 3 /var/dbb/v.1.0.6/
                                                                            3** Build start at 20200721.102817.028
     3 Invocation parameters:
                                                                                                                                                        > * /var/dbb/work github
                                                                            4** Build output located at /var/jenkins/g
      4 LIB.CICS.SOL
                                                                                                                                                        5 == Adding dbb-zappbuild-l/samples/Mortgag

6 == Writing build list file to /var/jenkin

7 == Throwing build scripts according to bu

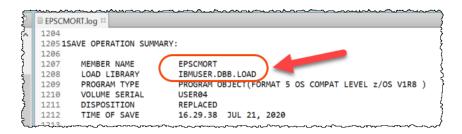
8 == Building files mapped to Cobol groovy
                                                                                                                                                          > 🗅 config
                  IGYOS4090-I The "LIB" option
                                                                                                                                                           < ⁵ gitispf
                                                                                                                                                               dbb-zappbuild-1
                                                                         9*** Building file dbb-zappbuild-1/samples
10** Writing build report data to /var/jenk
11** Writing build report to /var/jenkins/g
     9 Options in effect:
    10 NOADATA
                                                                                                                                                               BuildReport.html
           ADV
                                                                                                                                                               BuildReport.json
                                                                         12** Build ended at Tue Jul 21 22:29:38 GMT

13** Build State: CLEAN

14** Total files processed: 1

15** Total build time: 1 minutes, 21.570
           AFP(VOLATILE)
                                                                                                                                                               dbbub.log
           QUOTE
           ARCH(7)
                                                                                                                                                               ■ EPSCMORT Inc
        ARITH(COMPAT)
                                                                          16
                                                                                                                                                           > 🖹 slave.jar
        NOBLOCKO
                                                                          17 ** Build finished
           BUFSIZE(4096)
                                                                                                                                                             testrexx.rex
           CICS
                                                                                                                                                         /var/rocket
           CODEPAGE(1140)
                                                                                                                                                           /var/ienkins/gitispf
```

3.6. The z/OS PDS's will be populated:





4. Commit and Push the changes to Git Server

Once you are happy wit the results you can **commit** and **push** the changes to the Git Server.

4.1 In order to commit the changes you will use the commands:

Git Add - AD (to stage) and Git Commit and Push - CP.

This is similar as if you were using IDz.

Type **AD** (Git Add – to stage)

4.2 Type / to the file to stage

4.3 It shows. Press PF3

```
Command ===> _____ Scroll ===> <u>PAGE</u>

Add files to staging completed
```

4.4 CP for Git Commit and Push. Notice the Git Status is changed to "Staged"

```
qit@qithub.com:ReqiBrazil/dbb-zappbuild-1.qit
Git Repositoru
On Branch .....
                    sandbox
Pathname /var/jenkins/gitispf/dbb-zappbuild-1/samples/MortgageApplication/
    Filename
                                   Git Status
                                               File
    epscmort.cbl
<u>CP</u>
                                   Staged
    epscsmrd.cbl
                                               File
    epsmlist.cbl
                                               File
    epsmpmt.cbl
                                               File
    epsnbrvl.cbl
                                               File
 ****** Bottom of data *****
```

4.5 Type / to the file to commit and push

4.6 Press PF3

```
Command ===> Scroll ===> PAGE

ýsandbox 9e2a41f" Program was changed July 21 by Regi
Committer: IBMUSER <IBMUSER@cloud.provisioning.zos.ibm.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

Git config --global --edit

After doing this, you may fix the identity used for this commit
with:

git commit --amend --reset-author

1 file changed, 1 insertion(+), 1 deletion(-)
```

4.7 Try now the **ST** (Git Status)

```
git@github.com:RegiBrazil/dbb-zappbuild-1.git
Git Repository
On Branch ....:
                     sandbox
Pathname /var/jenkins/gitispf/dbb-zappbuild-1/samples/MortgageApplid
     Filename
                                     Git Status
                                                  Type
     epscmort.cbl
                                                  File
     epscsmrd.cbl
     epsmlist.cbl
                                                  File
     epsmpmt.cbl
     epsnbrvl.cbl
                   ********** Bottom of data ****
```

4.8 If now you go to *GitHub* you will see this code comitted/pushed there.

