Example using Git ISPF with https instead SSH

Created by rbarosa@us.ibm.com – Aug 06, 2020
Thanks to Liam Doherty for his help, I could not do without his guidance.

Introduction

This paper is created to help using **https** instead of **SSH** with Git ISPF. Few times the required port 22 used by SSH is blocked, while the https port usually is open. For a POC its easier using https.
If I try to clone using https on my system I have the error below:

```
IBMUSER:/SOW1/var/rocket: >cd /tmp
IBMUSER:/SOW1/tmp: >git clone https://github.com/IBM/dbb.git
Cloning into 'dbb'...
fatal: unable to access 'https://github.com/IBM/dbb.git/': SSL certificate probl
em: unable to get local issuer certificate
IBMUSER:/SOW1/tmp: >
 ===>
```

Rocket Git version 2.14.4 for z/OS supports the https protocol for access to remote repositories, in addition to the ssh protocol.
Some instructions below were extracted from Rocket PDF named "*Rocket-2.14.4-Release notes*".

Downloading a certificate bundle

The z/OS port of **Git 2.14.4** now supports the **https** Git remote access protocol in addition to the **ssh** protocol. To use this, you must set the environment variable GIT_SSL_CAINFO to point to a file containing the X.509 certificates of the public Certificate Authorities, in PEM format.

If you do not already have a suitable certificate file, you can download a current copy of the file from a trusted source and verify the signature of the file. A suggested source is the curl web site. If you have the Rocket ports of curl and open ssl installed, you can use the following commands. These assume that:

- The path of the directory in which Git and the related tools were installed is in the environment variable **RSUSR**.
- You have the write permission to that directory.
- You wish to store the certificate file in that directory.

Below the instructions from the Rocket PDF copied here:

Make sure that there is an "etc" subdirectory in the Rocket ported tools directory
```
mkdir -p $RSUSR/etc
cd $RSUSR/etc
```

Get the certificate file
```
curl -s -k https://curl.haxx.se/ca/cacert.pem -o cacert.pem
```

Get the signature file and extract just the hash
```
curl -s -k https://curl.haxx.se/ca/cacert.pem.sha256 | awk ' {print $1}' > cacert.pem.sha256
```

Generate the hash on the certificate file and compare it to the signaturefile.
If the signature matches, there will be no output from diff.
```
openssl dgst -sha256 cacert.pem | awk ' {print $2}' | diff - cacert.pem.sha256
```

See on my example:

```
IBMUSER:/u/ibmuser: >git --version
git version 2.14.4_zos_b09
IBMUSER:/u/ibmuser: >cd ..
IBMUSER:/u: >cd /var/rocket
IBMUSER:/SOW1/var/rocket: >mkdir -p /var/rocket/etc
IBMUSER:/SOW1/var/rocket: >cd /var/rocket/etc
IBMUSER:/SOW1/var/rocket/etc: >curl -s -k https://curl.haxx.se/ca/cacert.pem -o
cacert.pem
IBMUSER:/SOW1/var/rocket/etc: >curl -s -k https://curl.haxx.se/ca/cacert.pem.sha
256 | awk ' {print $1}' >cacert.pem.sha256
IBMUSER:/SOW1/var/rocket/etc: >openssl dgst -sha256 cacert.pem | awk ' {print $2
}' | diff -cacert.pem.sha256
diff: FSUM6001 Unknown option "-acert.pem.sha256"
Usage: diff Ý-BbefHhimNnrsw" Ý-C n" Ý-cÝn"" Ý-Difname" Ý-M mark" Ý-W optionÝ,opt
ion"..." path1 path2
WARNING: can't open config file: /rsusr/rocket/ssl/openssl.cnf
IBMUSER:/SOW1/var/rocket/etc: >
```

And I had this created:

```
IBMUSER:/SOW1/var/rocket/etc: >ls -al
total 512
drwxr-xr-x   2 IBMUSER  SYS1        8192 Aug  5 15:27 .
drwxrwxrwx  10 IBMUSER  SYS1        8192 Aug  5 15:24 ..
-rw-r--r--   1 IBMUSER  SYS1      222172 Aug  5 15:26 cacert.pem
-rw-r--r--   1 IBMUSER  SYS1          65 Aug  5 15:27 cacert.pem.sha256
IBMUSER:/SOW1/var/rocket/etc: >
```

Once this has been done, you can set `GIT_SSL_CAINFO` to point to the file:

```
export GIT_SSL_CAINFO=$RSUSR/etc/cacert.pem
```

It is also possible for a Git user to disable the certificate checking by entering the following command. This is not recommended.

```
git config --global http.sslVerify false
```

As I mentioned before, on my example when I try to clone using https I have the error below..

```
IBMUSER:/SOW1/var/rocket: >cd /tmp
IBMUSER:/SOW1/tmp: >git clone https://github.com/IBM/dbb.git
Cloning into 'dbb'...
fatal: unable to access 'https://github.com/IBM/dbb.git/': SSL certificate probl
em: unable to get local issuer certificate
IBMUSER:/SOW1/tmp: >
 ===>
```

I edited the ".profile" and added the line below (in yellow):

```
#!/bin/sh
#  changed to be Rocket v 2.14 - July 2020 - Regi
# Git env. vars.
export GIT_SHELL=/var/rocket/bin/bash
export GIT_EXEC_PATH=/var/rocket/v2.14/libexec/git-core
export GIT_TEMPLATE_DIR=/var/rocket/v2.14/share/git-core/templates
# Common env. vars
export PATH=$PATH:/var/rocket/v2.14/bin:/tmp/curl/bin:/u/ibmuser/CPM/gzip/bin
export LIBPATH=$LIBPATH:/tmp/curl/lib
export MANPATH=$MANPATH:/var/rocket/v2.14/man:/tmp/curl/share/man
export PERL5LIB=$PERL5LIB:/var/rocket/lib/perl5
export JAVA_HOME=/usr/lpp/java/J8.0_64
# ASCII support env. vars
export _CEE_RUNOPTS="FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)"
export _BPXK_AUTOCVT=ON
export _TAG_REDIR_ERR=txt
export _TAG_REDIR_IN=txt
export _TAG_REDIR_OUT=txt
export GIT_SSL_CAINFO=/var/rocket/etc/cacert.pem
```

Logged OFF – Logged ON..
And now the clone with https works.:

```
IBMUSER:/u/ibmuser: >cd /tmp
IBMUSER:/SOW1/tmp: >git clone https://github.com/IBM/dbb.git
Cloning into 'dbb'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 1464 (delta 7), reused 19 (delta 1), pack-reused 1434
Receiving objects: 100% (1464/1464), 1.91 MiB | 495.00 KiB/s, done.
Resolving deltas: 100% (807/807), done.
Checking out files: 100% (377/377), done.
IBMUSER:/SOW1/tmp: >
 ===>
```

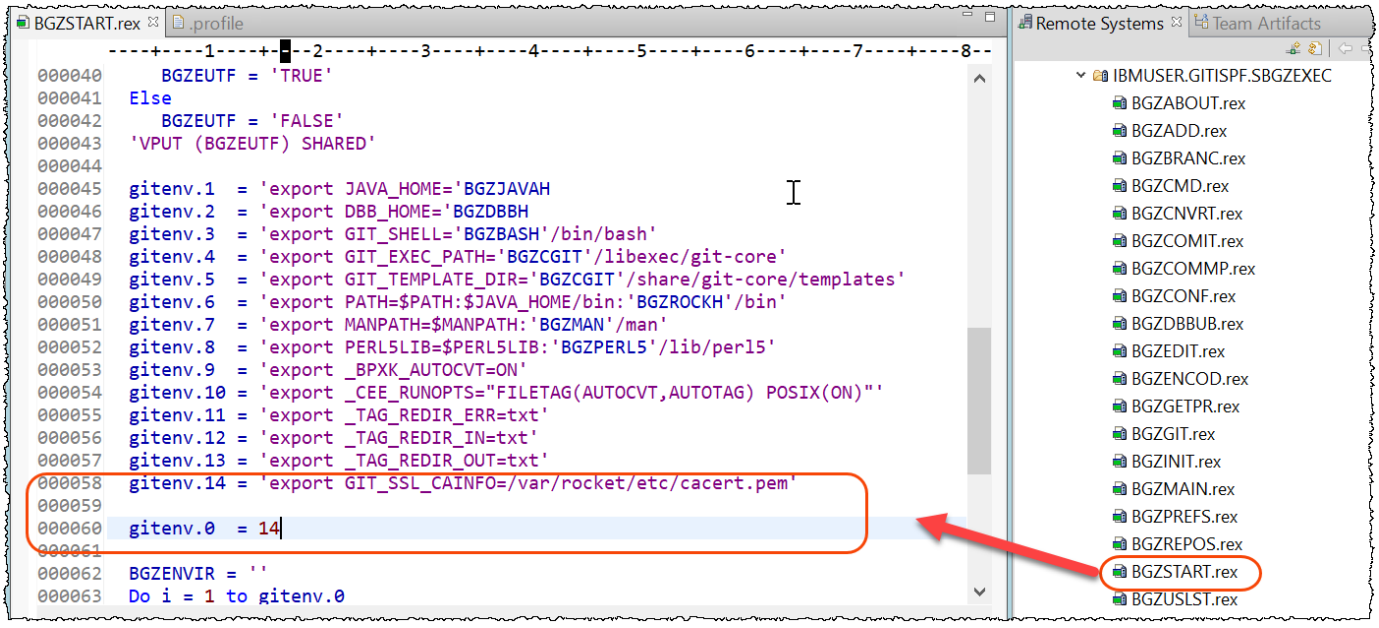
Updating the GitSipf panels

But it did not work in our ISPF client.
This is because ISPF is not using the **".profile"**.
You need to modify the REXX member **BGZSTART** to add that export:

```
gitenv.1  = 'export JAVA_HOME='BGZJAVAH
gitenv.2  = 'export DBB_HOME='BGZDBBH
gitenv.3  = 'export GIT_SHELL='BGZBASH'/bin/bash'
gitenv.4  = 'export GIT_EXEC_PATH='BGZCGIT'/libexec/git-core'
gitenv.5  = 'export GIT_TEMPLATE_DIR='BGZCGIT'/share/git-core/templates'
gitenv.6  = 'export PATH=$PATH:$JAVA_HOME/bin:'BGZROCKH'/bin'
gitenv.7  = 'export MANPATH=$MANPATH:'BGZMAN'/man'
gitenv.8  = 'export PERL5LIB=$PERL5LIB:'BGZPERL5'/lib/perl5'
gitenv.9  = 'export _BPXK_AUTOCVT=ON'
gitenv.10 = 'export _CEE_RUNOPTS="FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)"'
gitenv.11 = 'export _TAG_REDIR_ERR=txt'
gitenv.12 = 'export _TAG_REDIR_IN=txt'
gitenv.13 = 'export _TAG_REDIR_OUT=txt'
gitenv.14 = 'export GIT_SSL_CAINFO=/$RSUSR/etc/cacert.pem'
```

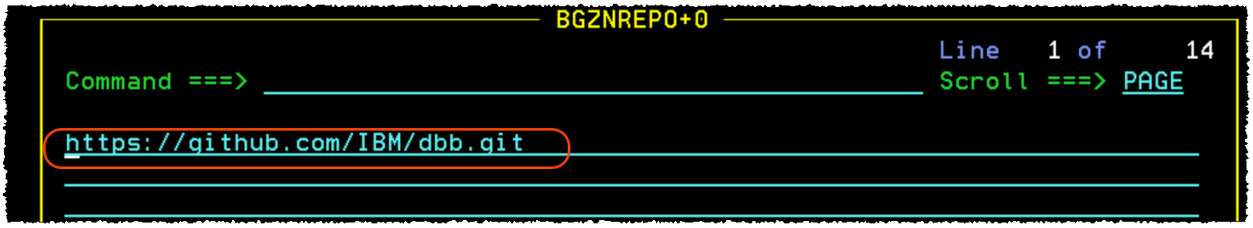
```
gitenv.0  = 14
```

On my example:

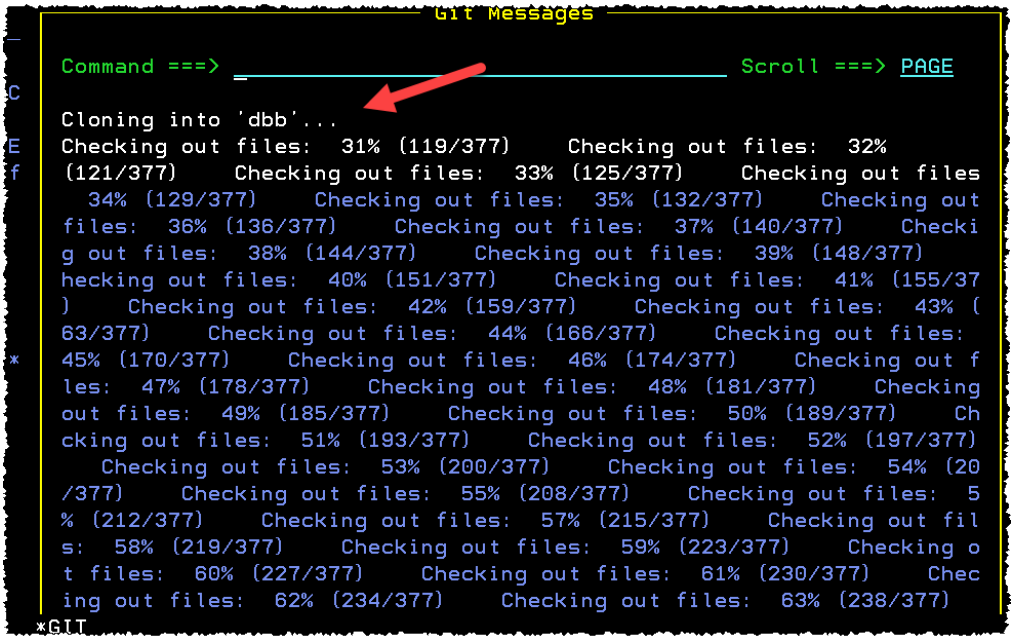


Now cloning with https:

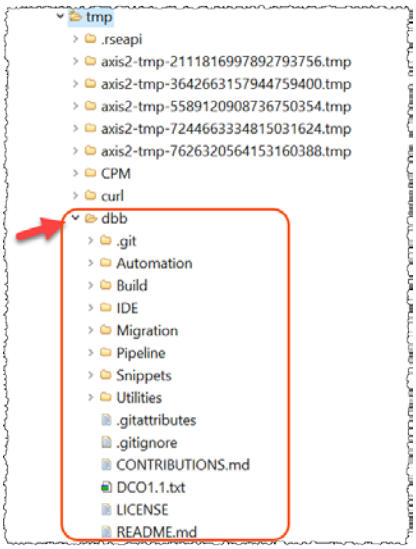
I used the public repo : https://github.com/IBM/dbb.git



And used **tmp** folder at USS.

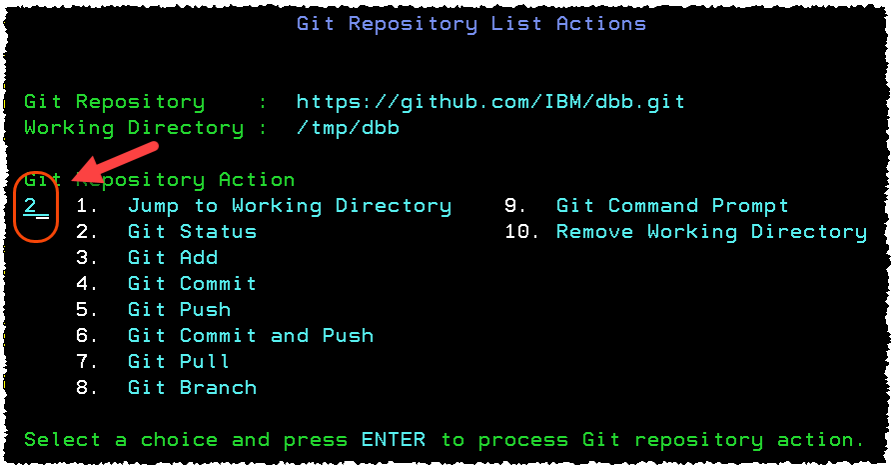


The Repo is cloned at USS:



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:

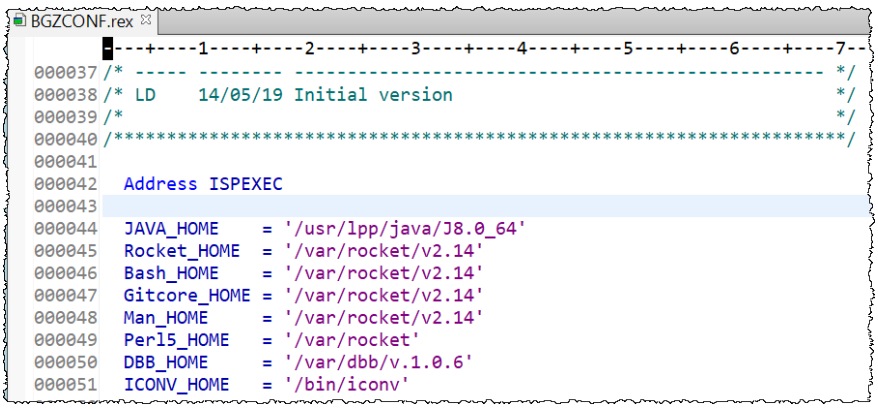


1.8 Result will be:

Troubles ?..

If you have errors like below

Be sure that your member *.SBGZEXEC(X) reflects the correct pointer. You may edit the ".profile" and compare if the values there are correct.



Some references that may be useful:

Interesting document:

https://w3.ibm.com/help/#/article/github_ent_ibm/github_security?requestedTopicId=github_security

Where it states:

If your computer natively supports SSH (e.g., Linux, OSX), then use SSH to push, not HTTPS. However, if you use Windows, pushing with HTTPS is sometimes preferred. You need a personal access token to get authenticated when pushing code. Refer to the following guides on the GitHub blog:

Creating a personal access token for the command line

(https://docs.github.com/en/github/authenticating-to-github/creating-a-personal-access-token)

We should add z/OS to that as we are Linux like.