IBM z/OS Connect EE V3.0

Customization - Basic Configuration
(1 of 2)

IBM Z
Wildfire Team –
Washington System Center

Lab Version Date: November 27, 2020

Table of Contents

Important: On the desktop there is a file named *Basic Configuration CopyPaste.txt*. This file contains commands and other text used in this workshop. Locate that file and open it. Use the copy-and-paste function (**Ctrl-C** and **Ctrl-V**) to enter commands or text. It will save time and help avoid typo errors. As a reminder text that appears in this file will be highlighted in yellow.

Mitch Johnson (mitchj@us.ibm.com)

General Exercise Information and Guidelines

- ✓ This exercise requires using z/OS user identity *USER1*. The password for this user will be provided by the lab instructor.
- ✓ Any time you have any questions about the use of IBM z/OS Explorer, 3270 screens, features or tools, do not hesitate to ask the instructor for assistance.
- ✓ Text in **bold** and highlighted in **yellow** in this document should be available for copying and pasting in a file named *Basic Configuration CopyPaste* file on the desktop.
- ✓ Please note that there may be minor differences between the screen shots in this exercise versus what you see when performing this exercise. These differences should not impact the completion of this exercise.

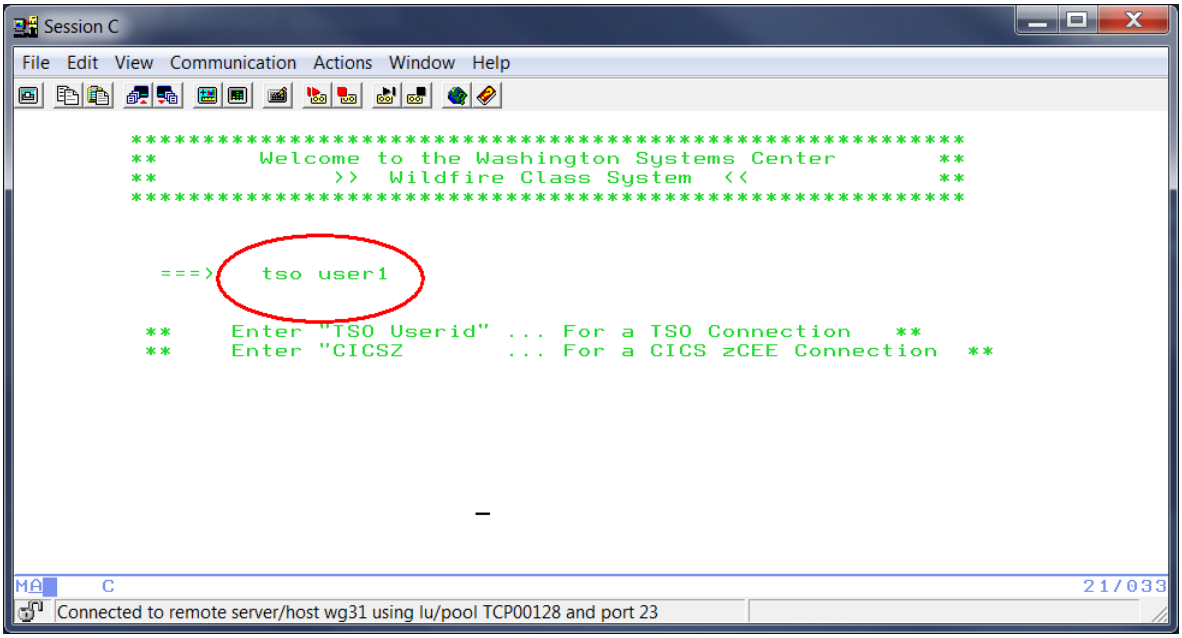
Mitch Johnson (mitchj@us.ibm.com)

Setup and Service Definitions

Run the jobs to setup RACF framework for server runtime

____ 1. Open the *WG31* icon on the workstation desktop. This will start a 3270-terminal session to your z/OS system.

> **Tech-Tip:** Desktop tools can be opened either by double clicking the icon or by selecting the icon and right mouse button clicking and then selecting the *Open* option.



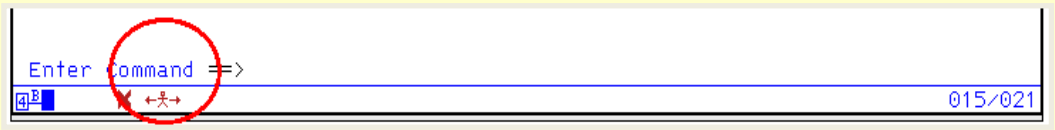
N.B., The 3270-terminal sessions and OMVS screen shots in the remainder of this and subsequent exercises are shown in reverse video simply for printing purposes

____ 2. Enter ***TSO USER1*** (see below) and press the 3270 **Enter** key (the **right-Ctrl** key sequence): The 3270-emulator used for this workshop (IBM Personal Communication) maps the 3270 enter key to the right **Ctrl** key (see below). Any references to the *Enter* key in non-3270 windows, OMVS terminal session, etc. refers to the key labeled *Enter* on the keyboard.

Mitch Johnson (mitchj@us.ibm.com)

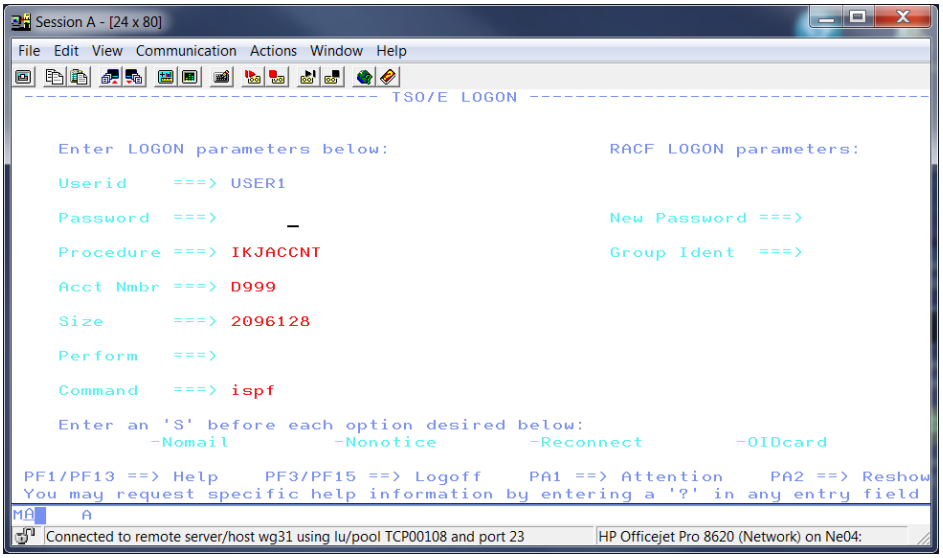
The instructions in this exercise will reference keyboard keys using a **bold** font. In the beginning explicit instructions regarding pressing keyboard keys will be provided. Eventually these explicit instructions to press the *Enter* key for example will not be included. If you are told to enter command then assume the appropriate *Enter* key should be pressed to have the command invoked or executed. Information which must be entered on a screen or panel will be in ***bold italics.*** References to text on a screen or panel will be in normal *italics*.

Tech-Tip: Different 3270-terminal emulators will display an icon similar to the *Personal Communications*®'s icon below at the bottom of the screen when the keyboard is locked. If this occurs use the left ***Ctrl*** key to reset or free the keyboard.



In this emulator the **Pause** key is mapped to the clear function. If your laptop has a **Pause** key use it to clear the screen. For newer laptops without a **Pause** key, use the key sequence **Fn-P** to clear the screen. If none of these works, try a **Break** key or an **Alt-C** key sequence.

____ 3. On the *TSO/E LOGON* panel, enter the password supplied by the instructors.

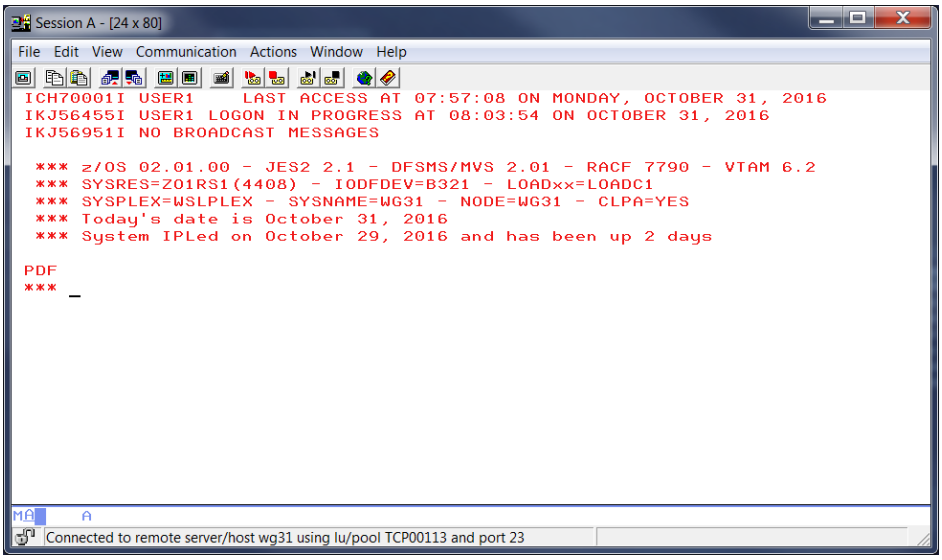


Tech-Tip: An **ISPF** or **PDF** command at the command prompt (===>) on this screen will automatically start ISPF. The copyright information displayed in Step 5 can be bypassed by entering ***ISPF NOLOGO*** or ***PDF NOLOGO*** at the command prompt.

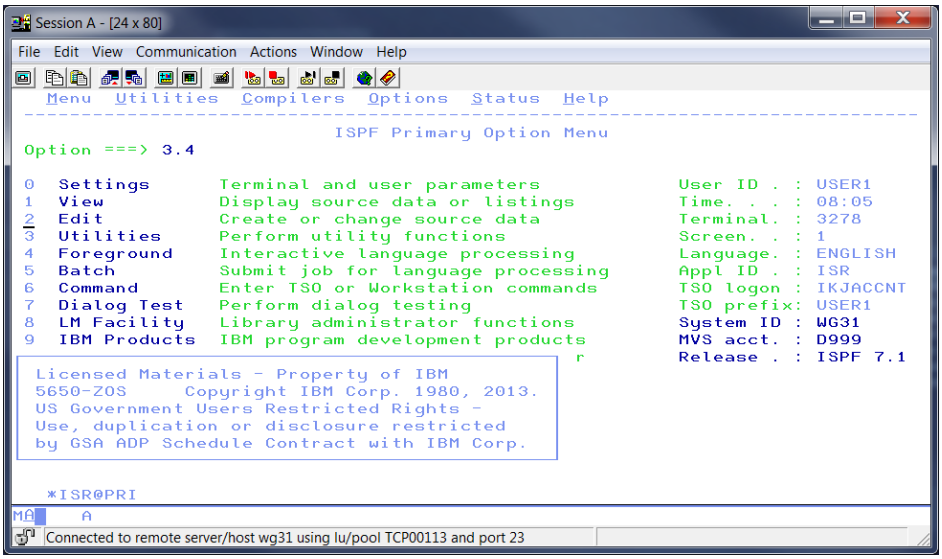
Tech-Tip: If for some reason you are disconnected from your TSO session and cannot log in because your TSO session is still active, you can enter an ***S*** beside *–Reconnect* near the bottom of the panel to reconnect to your session.

Mitch Johnson (mitchj@us.ibm.com)

____ 4. In a TSO session whenever you see the string ***** (three asterisk)** appear (as below) as the last line in any terminal output,there is more output is waiting to be displayed. Press the **Enter** key when you are ready to see this additional output. Also remember that notification messages such as jobs completing, etc. will not be displayed unless the **Enter** key is pressed



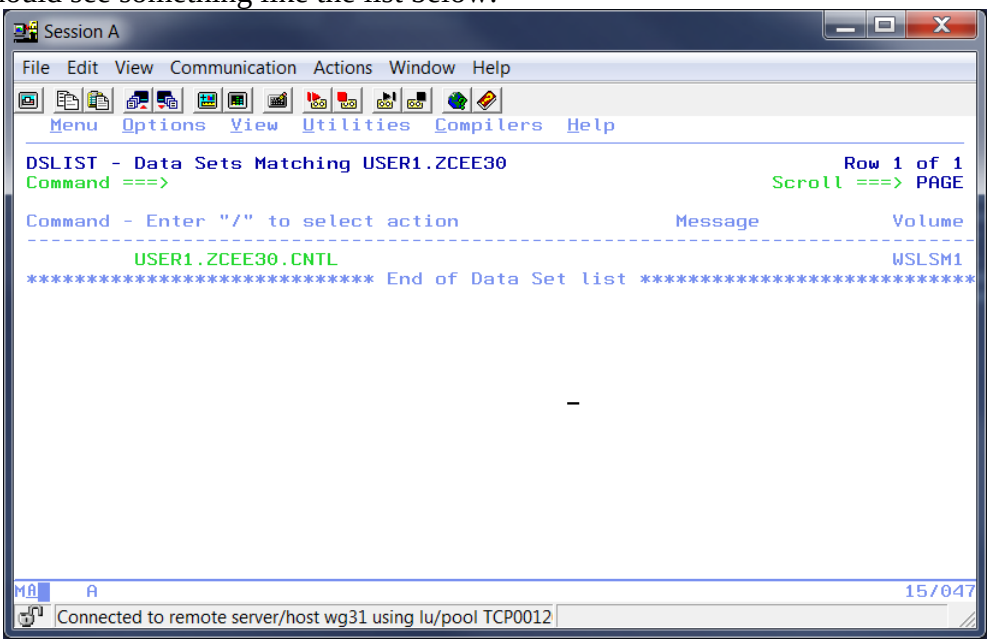
____ 5. You should now be at the main ISPF panel (see below). Press the **Enter** key to dismiss the *Copyright* statement.



____ 6. Enter ISPF command ***3.4*** in the area after the *Option*===> prompt in the upper left (see above) and press the **Enter** key to display the *Data Set List Utility* panel.

Mitch Johnson (mitchj@us.ibm.com)

____ 7. On the *Data Set List Utility* panel, enter ***USER1.ZCEE30*** in the area beside *DSNAME level*, and press the **Enter** key to display a list of data sets whose names begin with *USER1.ZCEE30*. You should see something like the list below:



____ 8. Enter ***E*** (for "edit") next to the data set *USER1.ZCEE30.CNTL*, and press **Enter**. You should see the a list of members in the partitioned data set.

Mitch Johnson (mitchj@us.ibm.com)

____ 9. Enter ***B*** (for "browse") next to member **ZCEERSTC,** and press **Enter**: You should see a job with several RACF commands (see below):

```
ADDGROUP LIBGRP OMVS(AUTOGID) OWNER(SYS1)

ADDUSER LIBANGE DFLTGRP(LIBGRP) OMVS(AUTOUID HOME(/u/libange/) -
  PROGRAM(/bin/sh)) NAME('LIBERTY ANGEL')  NOPASSWORD NOOIDCARD
ADDUSER LIBSERV DFLTGRP(LIBGRP) OMVS(AUTOUID HOME(/u/libserv/) -
  PROGRAM(/bin/sh)) NAME('LIBERTY SERVER')
ALTUSER LIBSERV PASSWORD(LIBSERV) NOEXPIRED

RDEFINE STARTED BAQSTRT.* UACC(NONE)    -
  STDATA(USER(LIBSERV) GROUP(LIBGRP) -
  PRIVILEGED(NO) TRUSTED(NO) TRACE(YES))
RDEFINE STARTED BAQZANGL.* UACC(NONE) -
  STDATA(USER(LIBANGE) GROUP(LIBGRP) -
  PRIVILEGED(NO) TRUSTED(NO) TRACE(YES))
SETROPTS RACLIST(STARTED) REFRESH

RDEFINE SURROGAT BPX.SRV.LIBSERV
RDEFINE SURROGAT LIBSERV.SUBMIT
PERMIT BPX.SRV.LIBSERV CLASS(SURROGAT) ID(ZCEEADM) ACC(READ)
PERMIT LIBSERV.SUBMIT  CLASS(SURROGAT) ID(ZCEEADM) ACC(READ)
SETROPTS RACLIST(SURROGAT) REFRESH
```

Tech-Tip: This job creates a RACF group and IDs for the Angel process and z/OS Connect EE V3.0 servers. It then it creates the two STARTED task profiles with these IDs assigned to these started tasks. Finally, the RACF resources that allows members of a z/OS Connect administrators group (ZCEEADM) to act as a surrogate of user LIBSERV and/or submit jobs as LIBSERV are defined.

____ 10. Enter command ***SUBMIT*** in the area after the command prompt (*Command ===>)* at the top of the screen and press the **Enter** key to submit this job for execution. You should get a message indicating the job has been submitted, along with the three asterisks indicating that additional output is being held before being displayed:

```
IKJ56250I JOB ZCEERSTC(JOB00060) SUBMITTED
***
```

____ 11. Press the **Enter** to display the additional output. You should then see either a message indicating the job has completed with MAXCC=0000 (which is good) or a redisplay of the ISPF browse panel. If the latter, keep pressing **Enter** until the job completes.

```
08.19.07 JOB00060 $HASP165 ZCEERSTC ENDED AT WG31  MAXCC=0000 CN(INTERNAL)
***
```

____ 12. Press the **F3** key to return to the list of members.

Mitch Johnson (mitchj@us.ibm.com)

____ 13. Enter ***B*** next to the **ZCEERSVR** member and press the **Enter** key. This is another set of RACF commands that define the RACF *SERVER* resources which allow the use of various z/OS authorized services. *WP102604 Getting Started Guide* describes these commands in more detail. ***Submit*** this job for execution (see Step 10 above). Allow this job to complete before continuing.

```
RDEFINE SERVER BBG.ANGEL UACC(NONE) OWNER(SYS1)
PERMIT  BBG.ANGEL CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM UACC(NONE) OWNER(SYS1)
PERMIT  BBG.AUTHMOD.BBGZSAFM -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.SAFCRED UACC(NONE)
PERMIT  BBG.AUTHMOD.BBGZSAFM.SAFCRED -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSWLM UACC(NONE)
PERMIT  BBG.AUTHMOD.BBGZSAFM.ZOSWLM  -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.TXRRS UACC(NONE)
PERMIT  BBG.AUTHMOD.BBGZSAFM.TXRRS   -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSDUMP UACC(NONE)
PERMIT  BBG.AUTHMOD.BBGZSAFM.ZOSDUMP  -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.SECPFX.BBGZDFLT UACC(NONE)
PERMIT  BBG.SECPFX.BBGZDFLT  -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.WOLA UACC(NONE) OWNER(SYS1)
PERMIT  BBG.AUTHMOD.BBGZSAFM.WOLA -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.LOCALCOM UACC(NONE) OWNER(SYS1)
PERMIT  BBG.AUTHMOD.BBGZSAFM.LOCALCOM -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSCFM UACC(NONE) OWNER(SYS1)
PERMIT  BBG.AUTHMOD.BBGZSCFM -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSCFM.WOLA UACC(NONE) OWNER(SYS1)
PERMIT  BBG.AUTHMOD.BBGZSCFM.WOLA -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
PERMIT  BBG.AUTHMOD.BBGZSCFM.WOLA -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.PRODMGR UACC(NONE) OWNER(SYS1)
PERMIT  BBG.AUTHMOD.BBGZSAFM.PRODMGR -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSAIO UACC(NONE) OWNER(SYS1)
PERMIT  BBG.AUTHMOD.BBGZSAFM.ZOSAIO -
      CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
SETROPTS RACLIST(SERVER) REFRESH
```

____ 14. Press **F3** to exit the browse session.

Summary

You just created a set of essential SAF profiles for z/OS Connect EE V3.0 to use. These are detailed in the *WP102604 Getting Started Guide*. The process was streamlined for this lab by coding them in a job so submitting this job would create what was needed.

Mitch Johnson (mitchj@us.ibm.com)

Create a z/OS Connect EE 3.0 server

z/OS Connect EE V3.0 was installed using SMP/E ahead of the workshop and the *zconsetup* command has been invoked to create a symbolic link to the */var/zosconnect/v3r0/extensions* directory. This directory contains property files which provide information required for locating z/OS Connect EE features and executables by Liberty at runtime.

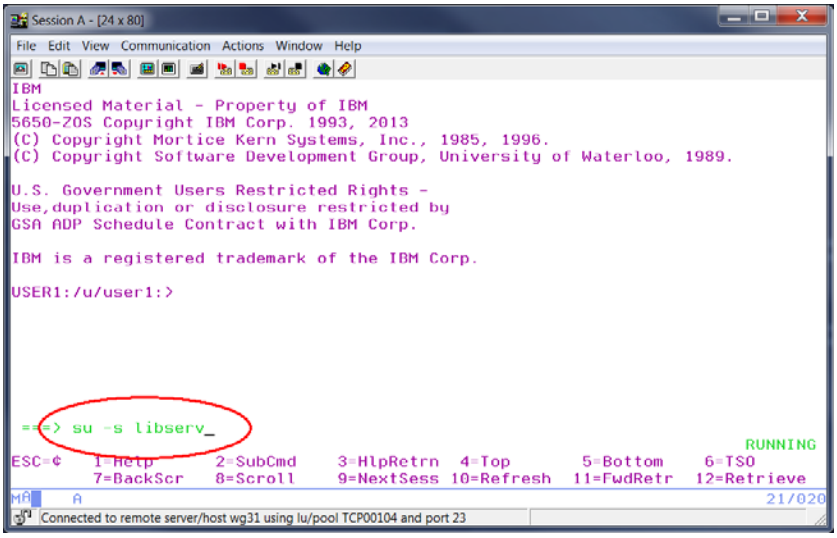
Now it's time to create a z/OS Connect server. This is done by executing a relatively simple shell script (*zosconnect*) provided with z/OS Connect EE. Remember that a z/OS Connect server runs in a Liberty runtime, so occasionally there will be references to Liberty.

Tech-Tip: The OMVS directories and files created by this script must be owned by the same RACF identity associated with the z/OS Connect started task (e.g. BAQSTRT). This is problematic since this RACF identity is restricted and cannot be used to logon to TSO or submit jobs to create these directories and files.

There are various ways to ensure this ownership is set correctly. One is to create the directories and files invoking the OMVS commands using your regular RACF identity. Then use the Unix System Services (USS) command *chown* (change owner) to change the ownership of the directories and files to the RACF identity of the started task. Another way is to use Telnet clients like PuTTY or TeraTerm to access OMVS using the restricted RACF identity and then invoking OMVS commands with the restricted identity.

This exercise uses a mixture of all these techniques. In a z/OS OMVS shell, the USS command *su* (switch user) is used to switch the current OMVS RACF identity to the RACF identity of the started task before invoking any commands, or it uses the *chown* command to change ownership when executing commands in a MVS batch job. PuTTY is used start an OMVS shell with the restricted identity to update the z/OS Connect server configuration file with USS *cp* (copy) commands. All of these techniques are mostly interchangeable and are shown to demonstrate the options available.

___1. In the existing 3270-terminal session, enter TSO command ***OMVS*** (e.g. ***TSO OMVS***) at the command prompt and press **Enter** to start an ISPF OMVS shell session. You should see the screen below.



Mitch Johnson (mitchj@us.ibm.com)

___2. At the prompt, enter OMVS command ***su –s libserv*** (see above) and press the 3270-terminal session's **Enter** key. Note that *libserv* is the RACF identity created and associated with the *BAQSTRT* started task when you ran the **ZCEESTC** job in the previous section.

> **Tech-Tip:** *LIBSERV* is the RACF identity created by the *ADDUSER* command executed when job ZCEESTC job was executed in the previous section. The *ALTUSER* command in the same job set the password for *LIBSERV* to *LIBSERV*.
>
> The ability to switch to this RACF identity without requiring a password in the *su* command was granted by defining the *BPX.SRV.LIBSERV* surrogate resource to RACF and then permitting user USER1 read access to this surrogate resource in *ZCEESTC*.
>
> It is very important that this command or alternatives be considered when creating a z/OS Connect EE (zCEE) server, otherwise the server may not start or perform correctly. The RACF identity associated with the zCEE started task, e.g. *LIBSERV* and must be either be the owner the configuration directories and files or explicitly given read/write access to them in *the /var/zosconnect/server/myServer* directory structure.

N.B. Instructions to press the **Enter** key will be omitted in subsequent steps; pressing **Enter** should be assumed whenever a command, script, etc. is to be executed.

___3. When you are logged on, enter the ***id*** command to confirm the user ID (*uid*) and group ID (*gid*) values are for LIBSERV and LIBGRP.

```
USER1:/u/user1:> su -s libserv
$ id
uid=200019(LIBSERV) gid=200017(LIBGRP)
$
```

___4. The z/OS Connect *zosconnect* shell script needs to be able to locate the Java executables. Use following command in the OMVS session to *export* the environment variable *JAVA_HOME*. This environment variable identifies the directory containing the location of the Java binaries.

```
export JAVA_HOME=/usr/lpp/java/J8.0_64
```

___5. Change to the directory containing the z/OS Connection script *zosconnect* using an OMVS *cd* command:

```
cd  /usr/lpp/IBM/zosconnect/v3r0/bin
```

> **Tech-Tip:** If you every have any questions about which directory your session is current using use the ***pwd*** (print working directory) to display the current directory path.

___6. Export environment variable (*WLP_USER_DIR*) to identify the directory where z/OS Connect EE V3.0 server configurations will reside.

```
export WLP_USER_DIR=/var/zosconnect
```

> **Tech-Tip:** The *WLP_USER_DIR* will be exported in the startup JCL. The value used in the JCL must be the same as the value used when the server was created.

Mitch Johnson (mitchj@us.ibm.com)

___7. Invoke the zosconnect script to create the z/OS Connect server named ***myServer:***

```
./zosconnect  create  myServer  --template=zosconnect:default
```

You should see the following in response.

```
Server myServer created.
```

Tech-Tip: Other templates which can be specified with the *zosconnect create* command are *zosconnect:apiRequester*, *zosconnect:sampleCicsIpicCatalogManager*, *sampleDb2ProjectManager*, *zosconnect:sampleImsDatabase*, *zosconnect:sampleImsPhonebook* , *sampleMqStockManager* and *zosconnect:sampleWolaCatalogManager*,

The best template to use to create a server is *zosconnect:apiRequester*. This template creates all resource sub directories. If the default template is used the same sub directories are not created.

Tech-Tip: Below is an example of creating a server using these same commands in an MVS batch job. Again, this job runs under the authority of RACF identity LIBSERV by using RACF SURROGAT resources. If a surrogate is not used, then the ownership of the directories and files created by running this job must be changed so they are owned by the user associated with the server's started task.

```
//ZCEESRVR JOB 'ZCEE Create',CLASS=A,REGION=0M,NOTIFY=&SYSUID,USER=LIBSERV
//*************************************************************
//*  SET SYMBOLS
//*************************************************************
//EXPORT EXPORT SYMLIST=(*)
// SET JAVAHOME='/usr/lpp/java/J8.0_64'
// SET ZCEEPATH='/usr/lpp/IBM/zosconnect/v3r0'
// SET SERVER='myServer'
// SET TEMPLATE='zosconnect:apiRequester'
// SET WLPUSER='/var/zosconnect'
// SET USER='LIBSERV'
// SET GROUP='LIBGRP'
//*************************************************************
//*  Step ZCEESRVR - Use the zosconnect command to create a server
//*************************************************************
//ZCEESRVR EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *,SYMBOLS=EXECSYS
 BPXBATCH SH +
   export JAVA_HOME=&JAVAHOME; +
   export WLP_USER_DIR=&WLPUSER; +
   &ZCEEPATH/bin/zosconnect create &SERVER +
   --template=&TEMPLATE
```


Mitch Johnson (mitchj@us.ibm.com)

___8. Enter the following OMVS command to display the contents of the */var/zosconnect/servers* subdirectory:

```
ls  -al  /var/zosconnect/servers
```

You should see:

```
$ ls -al /var/zosconnect/servers
total 96
drwxr-x--T   6 LIBSERV  SYS1        8192 Oct 31 11:57 .
drwxrwxrwx   4 BAGWELL  SYS1        8192 Oct 29 15:17 ..
drwxr-x--T   3 LIBSERV  SYS1        8192 Oct 29 15:17 .classCache
drwxr-x---   2 LIBSERV  SYS1        8192 Oct 31 11:57 .logs
drwxr-x---   3 LIBSERV  SYS1        8192 Oct 31 11:57 myServer
```

The *myServer* directory was created by *zosconnect* command in the previous step.

___9. Now display the contents of the */var/zosconnect/servers/myServer* directory by entering OMVS command:

```
ls  -al  /var/zosconnect/servers/myServer
```

You should see:

```
$ ls -al /var/zosconnect/servers/myServer
total 66
drwxr-x---   3 LIBSERV  SYS1        8192 Oct 31 11:57 .
drwxr-x--T   6 LIBSERV  SYS1        8192 Oct 31 11:57 ..
drwxr-x---   4 LIBSERV  SYS1        8192 Aug  1 16:23 resources
-rw-r-----   1 LIBSERV  SYS1          25 Oct 31 11:57 server.env
-rw-r-----   1 LIBSERV  SYS1         442 Oct 31 11:57 server.xml
drwxr-x---   3 LIBSERV  SYS1        8192 Oct 31 11:57 workarea
```

The "server.xml" file is the key configuration file for the Liberty z/OS server.

___10. Directory */wasetc/zcl3lab* contains xml files whose contents are Liberty and z/OS Connect configuration elements. This and subsequent exercises will have steps where these files will need to be included in the *server.xml* file in order to add features, functions, etc. There are advantages for accessing these *include files* directly from directory */var/zosconnect/servers/myServer* using a symbolic link. A symbolic link can be defined by entering this OMSV command.

```
ln -s /wasetc/zc3lab /var/zosconnect/servers/myServer/includes
```

A *ls -al* command would now show the existence of this symbolic link.

```
drwxr-x---   5 LIBSERV  SYS1        8192 Apr  2 09:58 .
drwxrwxrwx   4 JOHNSON  SYS1        8192 Apr  1 11:34 ..
lrwxrwxrwx   1 USER1    SYS1          14 Apr  2 09:58 includes -> /wasetc/zc3lab
drwxrwxrwx   4 LIBSERV  SYS1        8192 Apr  1 14:25 logs
drwxr-x---   4 LIBSERV  SYS1        8192 Apr  1 11:35 resources
-rw-r-----   1 LIBSERV  SYS1          67 Apr  1 11:34 server.env
-rw-r-----   1 LIBSERV  SYS1        2805 Apr  1 13:21 server.xml
drwxr-x---   5 LIBSERV  SYS1        8192 Apr  1 14:25 workarea
```


Mitch Johnson (mitchj@us.ibm.com)

___11. Terminate the OMVS session by entering the ***exit*** command twice to redisplay an ISPF panel. The first exit terminates the *libserv* session, and the second exit terminates the *user1* session.

Tech-Tip: Symbolic links have been created from a single directory to each of the server configuration directories on this image in order to simplify administration. Directory */var/zcee* was created and the command ***ln -s /var/zosconnect/servers/myServer /var/zcee/myServer*** was entered at an OMVS prompt to create this symbolic linkage for this server.
Start an OMVS shell and enter command ***ls -al /var/zcee*** at the command prompt. You should see something like what is shown below:

```
lrwxrwxrwx   1 USER1    SYS1          32 Jan 31 12:43 myServer ->
/var/zosconnect/servers/myServer
lrwxrwxrwx   1 JOHNSON  SYS1          36 Sep 10 08:40 wlpoidop ->
/var/ats/zosconnect/servers/wlpoidop
lrwxrwxrwx   1 JOHNSON  SYS1          36 Sep 10 08:39 wlpoidrp ->
/var/ats/zosconnect/servers/wlpoidrp
lrwxrwxrwx   1 JOHNSON  SYS1          14 Jan 15  2019 zc3lab -> /wasetc/zc3lab
lrwxrwxrwx   1 JOHNSON  SYS1           36 Jan 15  2019 zceeapir ->
/var/ats/zosconnect/servers/zceeapir
lrwxrwxrwx   1 JOHNSON  SYS1          39 Jan 15  2019 zceecics ->
var/cicsts/zosconnect/servers/zceecics
lrwxrwxrwx   1 JOHNSON  SYS1          25 Jan 15  2019 zceehats ->
/var/wlp/servers/zceehats
lrwxrwxrwx   1 JOHNSON  SYS1          36 Sep 12 11:14 zceeopid ->
/var/ats/zosconnect/servers/zceeopid
lrwxrwxrwx   1 JOHNSON  SYS1          36 Jan 15  2019 zceesrv1 ->
/var/ats/zosconnect/servers/zceesrv1
lrwxrwxrwx   1 JOHNSON  SYS1          36 Jan 15  2019 zceesrv2 ->
/var/ats/zosconnect/servers/zceesrv2
lrwxrwxrwx   1 JOHNSON  SYS1          36 Jul  2  2019 zceesrv3 ->
/var/ats/zosconnect/servers/zceesrv3
```

Notice that the use of symbolic links provides a single directory for accessing the different Liberty servers configured on this LPAR. These servers were configured with 4 different values for the WLP_USER_DIR environment variable, */var/zosconnect*, */var/ats/zosconnect*, */var/wlp* and */var/cicsts/zosconnect*. There is also a symbolic link to a common or shared directory */var/zcee/zc3lab* was also created for containing common configuration elements.

Summary

You just created a server, but right now it's just a skeleton without any service definitions or deployed APIs. Next you will customize the JCL start procedures for the Angel process and the server.

Mitch Johnson (mitchj@us.ibm.com)

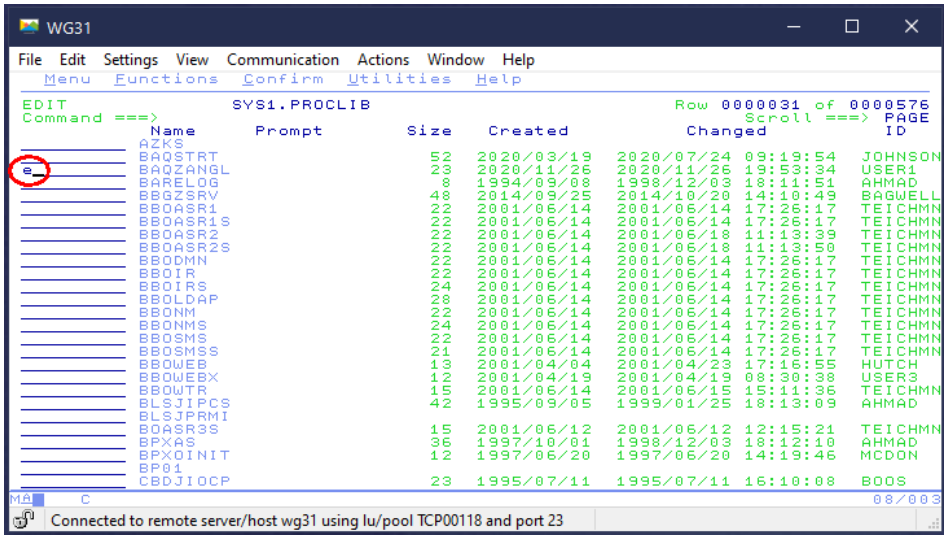
Customize the JCL start procedures and start the server

___1. In your 3270-terminal session, enter ISPF command **=3.4** in the command prompt area and press **Enter**.

___2. Enter ***SYS1.PROCLIB*** in the area beside *Dsname Level* and press **Enter.**

___3. Type an **E** (edit) next to the *SYS1.PROCLIB* data set and press **Enter**.

___4. You will now see a long list of members in the *SYS1.PROCLIB* data set. Enter command ***L BAQ*** after the command prompt (Command ===>) and press **Enter**. Locate the **BAQZANGL** member and place an **E** (edit) next to **BAQZANGL** to open this member in edit mode so changes can be made to the member.



This JCL procedure was copied this member to *SYS1.PROCLIB* prior to the workshop. This JCL is supplied in the z/OS Connect EE V3.0 SBAQSAMP target data set.

N.B. Instructions to press the **Enter** key should be assumed in subsequent steps as appropriate in order to execute commands, etc.

___5. Press **F3** to save the member and exit the edit session.

___6. Scroll up (**F7**) and locate member **BAQSTRT**. Place an **E** next to the member in order to open the member so changes can be made to the member.

This is the JCL procedure used to start the z/OS Connect EE V3.0 server. This JCL was copied to *SYS1.PROCLIB* prior to the workshop as well.

You are going to make three changes to this JCL procedure. The first is to replace the first line so the PARMS= specifies your server's name[1].

1 You could start the server with S BAQSTRT,PARMS='myServer' and avoid updating this line in the JCL. We chose to hard-code the server name for this workshop so the start is a simple S BAQSTRT.

Mitch Johnson (mitchj@us.ibm.com)

___7. Clear line 1 by placing your cursor at the start of the line (on the first slash in column 1):

```
000001 //BAQSTRT  PROC PARMS='defaultServer'
000002 //*
000003 //* Licensed Materials - Property of IBM
```

___8. Then press the **End** key. That should clear the line of all characters.

___9. Copy/Paste the JCL statement below on line 1 by using **Ctrl-C** of the text from the copy/paste file and then selecting *Edit* → *Paste* or **Ctrl-V** in the 3270-terminal session.

//BAQSTRT PROC PARMS='myServer'

The result should be:

```
000001 //BAQSTRT  PROC PARMS='myServer'
000002 //*
000003 //* Licensed Materials - Property of IBM
```

Important: In the above instructions the *Edit* → *Paste* translates to click the *Edit* tool in the tool bar of the 3270-terminal session and then select the *Paste* option. The → arrow will be used in later steps for selecting other tools in a tool bar and selecting an option in the 3270-terminal session and other windows.

You could have just overtyped *defaultServe*r with *myServer*. We chose to go with copy-and-paste to eliminate any chance of a case mis-match or typo. The field was cleared first because the original text in the line was longer than its replacement.

Mitch Johnson (mitchj@us.ibm.com)

___10. The next two lines to be updated are the *// SET ZCONHOME=* (line 38) and the *JAVA_HOME=* (line 47) environment variables. Do the following:

- Scroll down (**F8**) until you see the *SET ZCONHOME=* line:
- Place your cursor at the start of the line (the first slash):
- From the copy-and-paste file, copy the line:

```
// SET ZCONHOME='/usr/lpp/IBM/zosconnect/v3r0'
```

```
000037 //*
000038 // SET ZCONHOME='/usr/lpp/IBM/zosconnect/v3r0'
000039 //*
```

- Scroll down (**F8**) and locate the *JAVA_HOME=* line. It should be line 47:
- Place your cursor at the start of that line and press the **End** key to clear the line.
- From the copy-and-paste file, copy the line:

```
JAVA_HOME=/usr/lpp/java/J8.0_64
```

- Place your cursor at the start of that line and press the **End** key to clear the line.
- Locate the *WLP_USER_DIR=* line. It should be line 48:
- Place your cursor at the start of that line and press the **End** key to clear the line.
- From the copy-and-paste file, copy the line:

```
WLP_USER_DIR=/var/zosconnect
```

- In your 3270-terminal session, select *Edit* → *Paste* (you may need to remove a trailing > character). The result should be:

```
000045 //STDENV   DD   *
000046 _BPX_SHAREAS=YES
000047 JAVA_HOME=/usr/lpp/java/J8.0_64
000048 WLP_USER_DIR=/var/zosconnect
000049 #JVM_OPTIONS=<Optional JVM parameters>
```

Now you're ready to start the *Angel* process and the server.

Tech-Tip: An additional DD statement can be added to the JCL which will provide useful information regarding the status of the z/OS Connect. Normally detailed Liberty messages are written to an OMVS file in ASCII but if you add a DD statement for DD name MSGLOG and specify SYSOUT=* a subset of these messages will be included in the SPOOL of the task.

___11. Go to SDSF in your 3270-terminal session by entering ISPF command ***=sdsf.da*** in the command field and pressing **Enter.** The *da* stands for *display active*, and it is the z/OS facility for show running tasks and completed jobs.

___12. Next, enter ***PRE BAQ**** in the command field and press **Enter.** This sets a "prefix" so only tasks starting with string *BAQ* are shown. The Angel task starts with *BAQ*. After entering that command, you should see the new prefix is in effect.

Mitch Johnson (mitchj@us.ibm.com)

___14. Enter the MVS command ***S BAQZANGL*** after the command prompt.

> **Tech-Tip:** MVS and JES2 commands can be entered from SDSF by enter a / (slash) on the command line followed by the command itself (e.g. /D T). The command results can be found in the system log. If a command is especially long, then simply enter a / (slash) to display *a SDSF – System Command Extension* panel where a command can span multiple lines. When a MVS command must be entered, the instructions in these exercises will indicate that the command is a MVS command and you may enter the command at the prompt by using the / (slash) prefix or using the *SDSF – System Command Extension* panel.

___15. Keep pressing the **Enter** key until you should see the Angel process is running.

```
  Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF DA WG31     WG31     PAG  0  CPU   0              LINE 1-1 (1)
COMMAND INPUT ===>                                      SCROLL ===> CSR
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     BAQZANGL BAQZANGL STEP1    STC00045 LIBANGE    NS  FE  352   0.00   0.00
```

___16. You should see *LIBANGE* is the ID that owns the started task. That is a result of the RACF generated by the *ZCEESTC* job you ran earlier. The STARTED profile matched to the *BAQZANGL* start procedure and assigned the ID *LIBANGE* to this task.

```
  Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF DA WG31     WG31     PAG  0  CPU   0              LINE 1-1 (1)
COMMAND INPUT ===>                                      SCROLL ===> CSR
PREFIX=BAQ*  DEST=(ALL)  OWNER=*  SYSNAME=
NP   JOBNAME    CPU-Time ProcStep SR DP Pos C Owner    Status SysName  SPag SCP
     BAQZANGL       0.00 STEP1       FE NS    LIBANGE         WG31        0
```

___17. Change the prefix again by entering ***PRE BAQ**** at the command line. The *BAQZANGL* task no longer shows (it's still running, but the prefix no longer matches).

___18. Enter the MVS command ***S BAQSTRT*** after the command prompt and press **Enter** until you see an active BAQSTRT task.

```
  Display  Filter  View  Print  Options  Search  Help
-------------------------------------------------------------------------------
SDSF DA WG31     WG31     PAG  2  CPU   8              LINE 1-1 (1)
COMMAND INPUT ===>                                      SCROLL ===> CSR  =
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     BAQSTRT  BAQSTRT  ZCON     STC00048 LIBSERV    IN  F8  10T   0.00 21640
```

You should see that *LIBSERV* is the ID that owns the started task. That is a result of the RACF generated by the *ZCEESTC* job you ran earlier. The BAQSTRT started procedure matched STARTED profile resource and the task was assigned the ID *LIBSERV* to it.

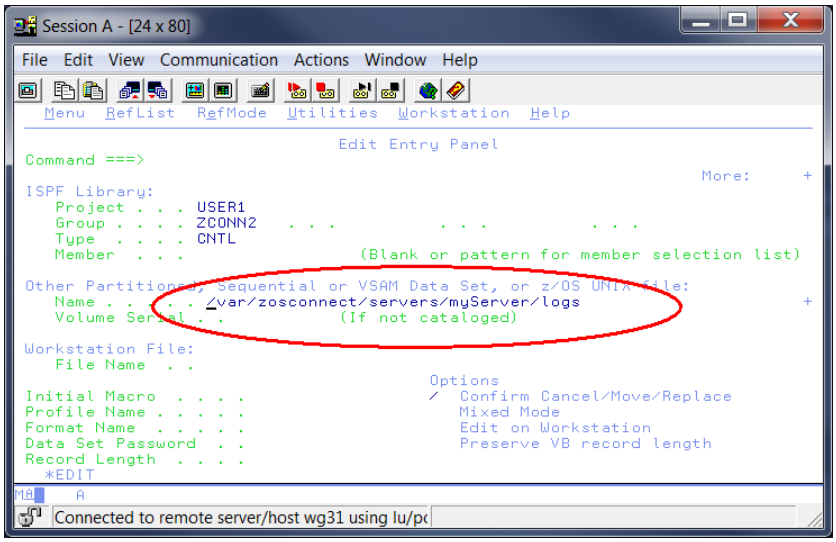
Note: The z/OS Connect EE server is running with *LIBSERV*'s RACF authority. This is why it was important to run the ***zosconnect create*** command as LIBSERV and why it is important in subsequence steps to use the *LIBSERV* identity when deploying Service ARhive (SAR) files. Also note that in later steps when files are included into the server.xml configuration file, the files that are included are owned by *LIBSERV*. All of this is done to avoid OMVS permission issues.

Both the Angel process and the z/OS Connect EE V3.0 server are started. Let's go look at the *messages.log* file for the server.

___20. Go to the ISPF Edit Entry Panel (option *2)* by entering ISPF ***=2*** on the command line and pressing **Enter**.

Tech-Tip: Most of the subsequent steps in this section can be performed using the IBM z/OS Explorer. If you are interested in doing these steps using this Eclipse tool, contact the instructor.

___21. Enter ***/var/zosconnect/servers/myServer/logs*** into the area beside *Name* under *Other Partitioned, Sequential or VSAM Data Set, or z/OS UNIX file:* and press **Enter**.



Tech-Tip: Try using the symbolic link discussed earlier, e.g. ***/var/zcee/myServer/logs*** instead.

Mitch Johnson (mitchj@us.ibm.com)

___22. This will display the contents of the OMVS directory where the z/OS Connection messages are written to file *messages.log*.

```
------------------------------------------------------------------------------
                          z/OS UNIX Directory List            Row 1 to 4 of 4
Command ===>                                                 Scroll ===> PAGE
Time zone EST5EDT is used to calculate the displayed date and time values.
Pathname . : /SYSTEM/var/zosconnect/servers/myServer/logs
EUID . . . : 8470391
Command  Filename                      Message          Type Permission
------------------------------------------------------------------------------
         .                                              Dir  rwxrwxrwx
         ..                                             Dir  rwxr-x---
         messages.log                                   File rw-rw-rw-
         state                                          Dir  rwxrwxrwx
****************************** Bottom of data ********************************
```

___23. Messages are written in ASCII so to view them in a 3270-terminal session use the ***VA*** (View ASCII) line command. Enter line command ***VA*** beside *messages.log* and press **Enter** twice.

___24. You should see the *View* panel open and the *messages.log* file contents displayed in EBCDIC:

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
VIEW       /SYSTEM/var/zosconnect/servers/myServer/logs/me Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
****** ***************************** Top of Data ******************************
000001 ********************************************************************************
000002 product = WAS FOR Z/OS 18.0.0.3, z/OS Connect 03.00.15 (wlp-1.0.22.cl180
000003 wlp.install.dir = /shared/IBM/zosconnect/v3r0/wlp/
000004 server.config.dir = /var/zosconnect/servers/myServer/
000005 java.home = /shared/java/J8.0_64
000006 java.version = 1.8.0
000007 java.runtime = Java(TM) SE Runtime Environment (pmz6480sr3fp20-20161019_
000008 os = z/OS (02.01.00; s390x) (en_US)
000009 process = 16777379@WG31
000010 ********************************************************************************
000011 [11/29/18 19:49:24:632 GMT] 0000000d com.ibm.ws.kernel.launch.internal.F
000012 [11/29/18 19:49:26:483 GMT] 0000001b com.ibm.ws.config.xml.internal.XMLC
000013 [11/29/18 19:49:26:527 GMT] 0000001b com.ibm.ws.config.xml.internal.XMLC
000014 [11/29/18 19:49:26:529 GMT] 0000001b com.ibm.ws.config.xml.internal.XMLC
000015 [11/29/18 19:49:26:530 GMT] 0000001b com.ibm.ws.config.xml.internal.XMLC
000016 [11/29/18 19:49:26:532 GMT] 0000001b com.ibm.ws.config.xml.internal.XMLC
000017 [11/29/18 19:49:26:777 GMT] 00000024 com.ibm.ws.logging.internal.TraceSp
000018 [11/29/18 19:49:27:228 GMT] 0000001b com.ibm.ws.zos.core.internal.Native
000019 [11/29/18 19:49:27:230 GMT] 0000001b com.ibm.ws.zos.core.internal.Native
000020 [11/29/18 19:49:27:244 GMT] 0000001b com.ibm.ws.zos.core.internal.Native
```

___25. You need to scroll to the right to see more message details. Enter ***100*** after the command prompt (===>) and press the **F11** key.

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
--------------------------------------------------------------------------------
VIEW       /SYSTEM/var/zosconnect/servers/myServer/logs/me Columns 00101 00172
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************

0.22.cl180320180905-2337)









WKE0001I: The server myServer has been launched.
WKB0103I: Authorized service group KERNEL is available.
WKB0103I: Authorized service group LOCALCOM is available.
WKB0103I: Authorized service group PRODMGR is available.
WKB0103I: Authorized service group SAFCRED is available.
WKB0103I: Authorized service group TXRRS is available.
WKB0103I: Authorized service group WOLA is available.
WKB0103I: Authorized service group ZOSAIO is available.
WKB0103I: Authorized service group ZOSDUMP is available.    .
```

Note that the **F8** key can be used to scroll forward and the **F7** key can be used to scroll backward. **F10** can be used to scroll leftward and F**11** key can be used to scroll rightward. A numeric value on the command prompt will be used to determine the number of lines to scroll forward or backward or the number of columns for scrolling leftward or rightward.

Mitch Johnson (mitchj@us.ibm.com)

___26. Look at the messages and note the following:

```
000012 WKB0103I: Authorized service group KERNEL is available.
000013 WKB0103I: Authorized service group LOCALCOM is available.
000014 WKB0103I: Authorized service group PRODMGR is available.
000015 WKB0103I: Authorized service group SAFCRED is available.
000016 WKB0103I: Authorized service group TXRRS is available.
000017 WKB0103I: Authorized service group WOLA is available.
000018 WKB0103I: Authorized service group ZOSAIO is available.
000019 WKB0103I: Authorized service group ZOSDUMP is available.
000020 WKB0103I: Authorized service group ZOSWLM is available.
000021 WKB0103I: Authorized service group CLIENT.WOLA is available.
000022 WKB0108I: IBM Corp product z/OS Connect version 03.00 successfully regis
000023 WKB0112I: The number of successfully registered products with z/OS is 1.
000024 WKE0002I: The kernel started after 3.05 seconds
000025 WKF0007I: Feature update started.
000026 WKS0007I: The security service is starting...
000027 NA1001I: WebSphere Dynamic Cache instance named baseCache initialized su
000028 NA1071I: The cache provider default is being used.
000029 NA1056I: Dynamic Cache (object cache) initialized successfully.
000030 WKO0229I: Native Asynchronous I/O support for z/OS has been activated.
000031 WKS4103I: Creating the LTPA keys. This may take a few seconds.
000032 WKS1123I: The collective authentication plugin with class name NullColle
000033 QR0000I: z/OS Connect Enterprise Edition version 3.0.15. (20181120-1404)
000034 WKO0219I: TCP Channel defaultHttpEndpoint has been started and is now li
000035 WKS4104A: LTPA keys created in 1.513 seconds. LTPA key file: /var/zoscon
000036 WKS4105I: LTPA configuration is ready after 1.527 seconds.
000037 WKF0015I: The server has the following interim fixes active in the runti
000038 WKF0012I: The server installed the following features: [servlet-3.1, ssl
000039 WKF0008I: Feature update completed in 5.539 seconds.
000040 WKF0011I: The server myServer is ready to run a smarter planet.
000041 VE0169I: Loading Web Module: z/OS Connect.
000042 VE0250I: Web Module z/OS Connect has been bound to default_host.
000043 WKT0016I: Web application available (default_host): http://wg31.washingt
000044 SN8501I: The session manager did not find a persistent storage location;
000045 SN0176I: A new session context will be created for application key defau
000046 SN0172I: The session manager is using the Java default SecureRandom impl
000047 NA1056I: Dynamic Cache (object cache) initialized successfully.
000048 VE0242I: [com.ibm.zosconnect] [/] [com.ibm.zosconnect.internal.web.Servi
000049 WKS9122I:  For URL /* in application com.ibm.zosconnect, the following H
```

1. The "Authorized service …. is available" messages indicate the SERVER profiles (**ZCEESAF** job) are in effect for this server and the ID under which it runs. Of particular interest is SAFCRED (for later when we use SAF for security).
2. Version 3.0.15.0 (20181120-1404) indicates the maintenance made available in November 20th of 2018 is in effect. The values you see may be different.
3. The server is started and ready to run.

Mitch Johnson (mitchj@us.ibm.com)

___25. Use the **F3** key to go back to the *Edit Entry Panel* and use the space bar or **Backspace** key to remove the */logs* subdirectory from the directory name field. Press **Enter** to display the contents of */var/zosconnect/servers/myServer*. You should see:

```
   Menu  Utilities  View  Options  Help
--------------------------------------------------------------------------------
                           z/OS UNIX Directory List            Row 1 to 8 of 8
 Command ===>                                                 Scroll ===> PAGE
 Time zone EST5EDT is used to calculate the displayed date and time values.
 Pathname . : /SYSTEM/var/zosconnect/servers/myServer
 EUID . . . : 8470391
 Command  Filename                      Message          Type Permission
 ------------------------------------------------------------------------------
          .                                              Dir  rwxr-x---
          ..                                             Dir  rwxr-x--T
          logs                                           Dir  rwxrwxrwx
          resources                                      Dir  rwxrwxrwx
          server.env                                     File rw-r-----
          server.xml                                     File rw-r-----
          workarea                                       Dir  rwxr-x---
 ****************************** Bottom of data ********************************
```

___26. Use the ***VA*** (View ASCII) line command to open the *server.xml* file in browse mode. You should see:

```
000001 <?xml version="1.0" encoding="UTF-8"?>
000002 <server description="new server">
000003
000004     <!-- Enable features -->
000005     <featureManager>
000006         <feature>zosconnect:zosConnect-2.0</feature>
000007         <feature>zosconnect:zosConnectCommands-1.0</feature>
000008     </featureManager>
000009
000010     <!-- To access this server from a remote client add a host attribute
000011     <httpEndpoint id="defaultHttpEndpoint"
000012                   host="*"
000013                   httpPort="9080"
000014                   httpsPort="9443" />
000015
000016     <!-- add cors to allow cross origin access, e.g. when using swagger
000017     <cors id="defaultCORSConfig"
000018           domain="/"
000019           allowedOrigins="*"
000020           allowedMethods="GET, POST, PUT, DELETE, OPTIONS"
000021           allowedHeaders="Origin, Content-Type, Authorization"
000022           allowCredentials="true"
000023           maxAge="3600"/>
000024
000025  <!-- NOTE: Disabling automatic polling for changes to configuration fil
000026       deployed services and APIs is a prudent option for z/OS Connect EE
000027    Polling might be convenient for iterative development and test
000028    systems, but not for production.
000029
000030       Configuration elements that can drive significant polling activity
000031    default are specified below to explicitly disable automatic polling.
000032    Further element types to consider for polling interval include
000033    zosconnect_zosConnectDataXform (default 2 seconds) and
000034    keyStore (default 500 milliseconds).
```


Mitch Johnson (mitchj@us.ibm.com)

```
000035
000036    Consider setting the updateTrigger attribute to "polled" if changes
000037    to associated resources need to be picked up automatically, and tune
000038    the polling interval accordingly. The attribute that controls polling
000039    frequency for each of these elements is included, together with its
000040    associated default value.
000041     -->
000042
000043  <!-- config requires updateTrigger="mbean" for REFRESH command support
000044  <config updateTrigger="mbean" monitorInterval="500"/>
000045
000046     <!-- zosConnect APIs -->
000047     <zosconnect_zosConnectAPIs updateTrigger="disabled" pollingRate="5s"
000048
000049     <!-- zosConnect Services -->
000050  <zosconnect_services updateTrigger="disabled" pollingRate="5s"/>
000051
000052  <!-- applicationMonitor is not applicable for zCEE servers -->
000053  <applicationMonitor updateTrigger="disabled" dropinsEnabled="false"/>
000054
000055 </server>
```

That is a *very* minimal server.xml file created from the default template. Other templates will generate server.xml files based on their parameters.
In the next step we will have you copy in a new server.xml file that will add quite a few more XML elements. The dynamic nature of Liberty will incorporate the new configuration without requiring a server restart.

Tech-Tip: Note that by default the *updateTrigger* attributes are disabled for APIs and services. These will need to be enabled if auto update is desired.

___27. Add the following include after the ***<server description=<new server”>*** line.

<include location="${server.config.dir}/includes/basicSecurity.xml"/>

Tech-Tip: This include element is taking advantage of an environment variable (*server.config.dir*) whose value is set to the directory containing the *server.xml* file used to initialize this instance of a Liberty server. The *includes* subdirectory is the symbolic link that references directory */wasetc/zc3lab* (the location of the include files shared among multiple servers. The value of using environment variables and symbolic links in the server.xml makes this configuration element reusable across multiple instances of servers and multiple OMVS directory configurations.

___28. Including this file (see below) adds a new feature to the server, *appSecurity-2.0*, which enables application security and some of the other elements required for basic security.

```
<server description="basic security">

    <!-- Enable features -->
    <featureManager>
        <feature>appSecurity-2.0</feature>
    </featureManager>

    <keyStore id="defaultKeyStore" password="Liberty"/>

    <webAppSecurity allowFailOverToBasicAuth="true" />

    <basicRegistry id="basic1" realm="zosConnect">
        <user name="Fred" password="fredpwd" />
    </basicRegistry>

    <authorization-roles id="zos.connect.access.roles">
        <security-role name="zosConnectAccess">
             <user name="Fred"/>
         </security-role>
     </authorization-roles>

</server>
```

Figure 1: Contents of basicSecurity.xml

"Basic" means a user is authenticated by providing a user identity and password. In this case the user registry will be provided by the Liberty server itself. For now, it's a simple way to satisfy the security requirements of z/OS Connect EE V3.0. In subsequent section of this exercise, SAF (e.g. RACF) will be implemented.

___29. Use the **F3** key to end the edit session.

___30. Adding this file using an *include* statement refreshes the z/OS Connect EE V3.0 server. Enter the following MVS command to refresh the configuration ***F BAQSTRT,ZCON,REFRESH***

Next, we're going to do a preliminary test of your z/OS Connect EE V30 server, even though you do not yet have any services or APIs defined.

> **Tech-Tip:** MVS commands can be entered using SDSF. In ISPF, enter the command ***=SDSF.LOG*** and add the / (slash) prefix to the command, e.g. /F BAQSTRT,ZCON,REFRESH

___31. When the message that the server configuration has been successfully updated appears in the *messages.log* file (see below), open the Firefox browser on your desktop.

```
CWPKI0803A: SSL certificate created in 5.313 seconds. SSL key file: /var/zosconnect/servers/myServer/resources/security/key.jks
CWWKS9112A: The web application security settings have changed. The following properties were modified: allowFailOverToBasicAuth=true
CWWKS9120I: Authorization roles with id="zos.connect.access.roles" have been successfully processed
CWWKG0017I: The server configuration was successfully updated in 5.481 seconds.
CWWKO0219I: TCP Channel defaultHttpEndpoint-ssl has been started and is now listening for requests on host *  (IPv4) port 9443.
```


Mitch Johnson (mitchj@us.ibm.com)

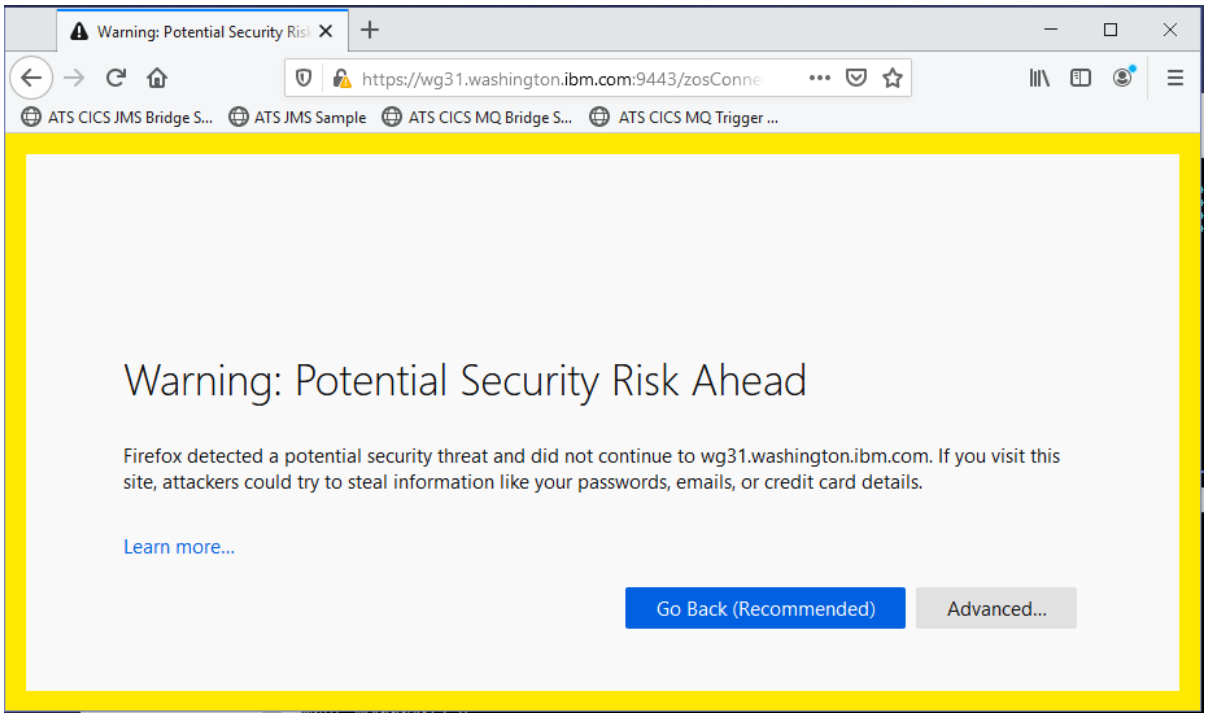
Verify the z/OS Connect server configuration

___1. Enter the following as the URL:

https://wg31.washington.ibm.com:9443/zosConnect/apis

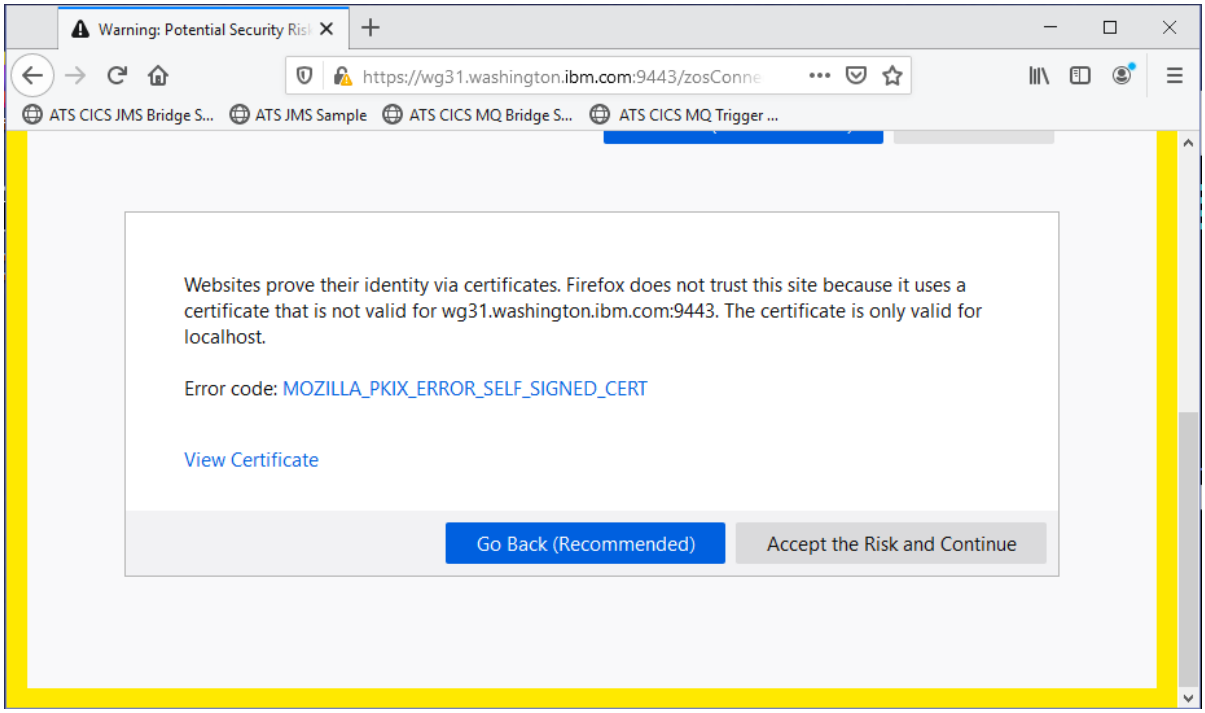
This will query the server for configured APIs. Even though you will receive a response indicating no APIs are configured, this is still a good test of connectivity to the server. But before we see the screen showing no APIs are defined, we first must address an issue where the z/OS Connect EE server is using a self-signed certificate that is not recognized by the browser, and we must authenticate to the z/OS Connect EE server with a user identity and password.

___2. Initially you will be challenged by Firefox because the digital certificate used by the Liberty z/OS server is self-signed (recall that we're using simple basic security for the time being). Click on the **Advanced** button to continue.



Tech-Tip: It is very important to access the z/OS Connect server from a browser prior to any testing using the Swagger UI. Accessing a z/OS Connect URL from a browser starts an SSL handshake between the browser and the server. If this handshake has not performed prior to performing any test, the test will fail with no message in the browser and no explanation. Ensuring this handshake has been performed is why you may be directed to access a z/OS Connect URL prior to using the Swagger UI during this exercise.

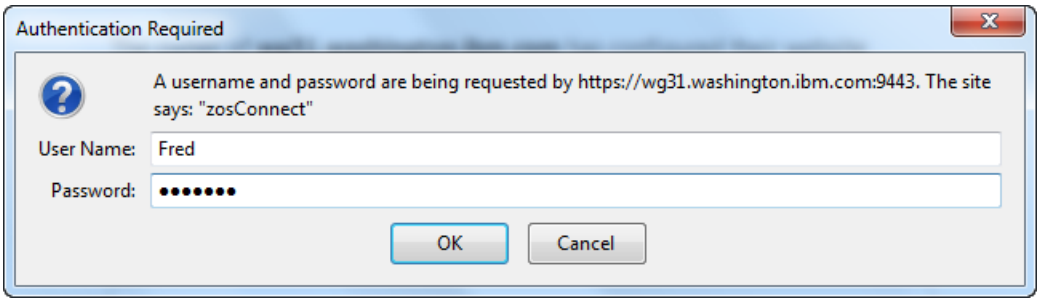
___3. Scroll down and click the **Accept the Risk and Continue** button to continue.



Tech-Tip: The configuration of this Firefox browser has been configured to accept self-signed certificates and to not permanently save certificates under these circumstances. This means that these security risk screens will be displayed every time a browser is restarted until a permanent certificate is installed. This action was enabled by changing the Firefox security.certerrors.permanentOverride preference to false.

Mitch Johnson (mitchj@us.ibm.com)

___4. Next you will see a prompt you for a userid and password:



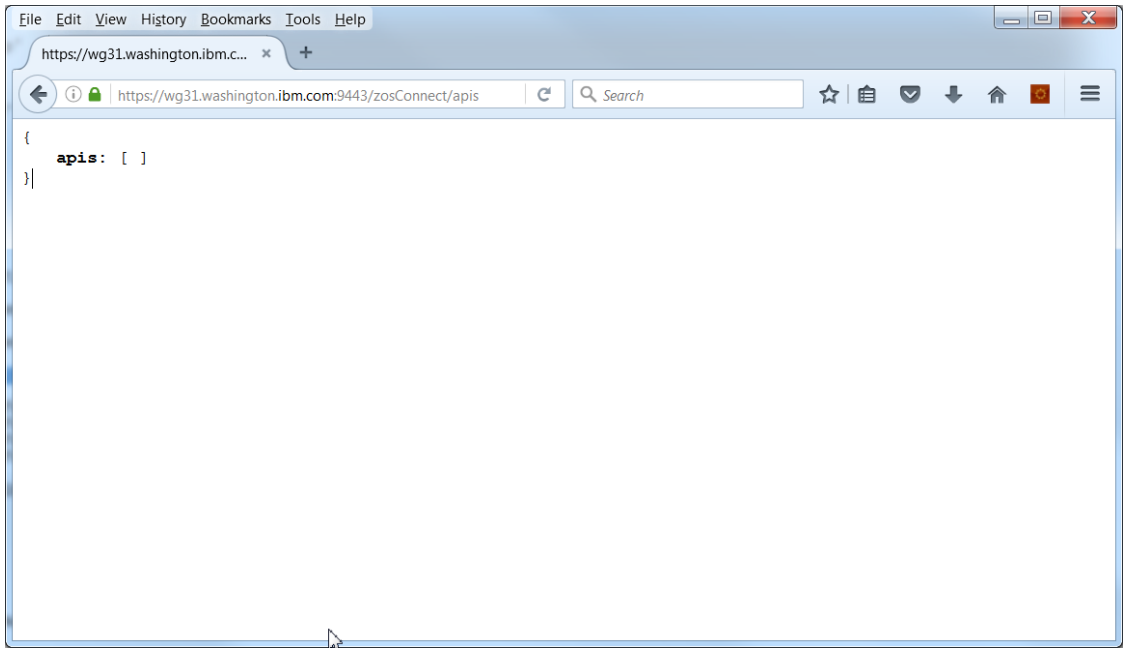
Enter a *User name* of ***Fred*** and a *Password* of **fredpwd** (case matters) and click **OK**.

Note: The ID and password is defined in the basicSecurity.xml file:

```
<basicRegistry id="basic1" realm="zosConnect">
   <user name="Fred" password="fredpwd" />
</basicRegistry>
```

Our objective for security at this point in the workshop is simplicity. That's why we're using the security elements coded in the server.xml. Later we will use RACF.

___5. You should now see the following. That is the expected result since no APIs have been installed.



___6. Enter the following as the URL:

https://wg31.washington.ibm.com:9443/zosConnect/services

That will query the server for configured services. You should also get a response indicating no services are configured. This is expected.

___7. An alternative to using a browser, you can verify the server by using the cURL command in a *Command Prompt* session.

curl -X GET --user Fred:fredpwd --header "Content-Type: application/json" --insecure https://wg31.washington.ibm.com:9443/zosConnect/apis

with results:

{"apis":[]}

curl -X GET --user Fred:fredpwd --header "Content-Type: application/json" --insecure https://wg31.washington.ibm.com:9443/zosConnect/services

with results:

{"zosConnectServices":[]}

Summary

You customized the JCL start procedures and started both the Angel process and the server. You have added basic security for z/OS Connect EE V3.0. Finally, you verified basic operations with the browser and optionally the cURL command.

Mitch Johnson (mitchj@us.ibm.com)

Deploying Services and APIs

In this section, you will deploy previously generated service archive and API archives files using z/OS Connect EE RESTful administrative interfaces. But before the services can be deployed, the features required by the services must be installed in the running server.

Update the z/OS Connect EE Server Configuration

Before Services and APIs for the application can be deployed, the *server.xml* needs additional configuration information.

___1. Edit the *server.xml* in */var/zosconnect/servers/myServer* and add the following *include* after the ***<server description=<new server”>*** line.

<include location="${server.config.dir}/includes/ipic.xml"/>

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
<include location="/wasetc/zc3lab/basicSecurity.xml"/>
<include location="/wasetc/zc3lab/ipic.xml"/>
```

Tech-Tip: If a feature required by a service is not currently installed in the server, the deployment of the service will fail.

___2. Including this file adds a new feature to the server, *cicsService-1.0*, which enables support for connecting to CICS regions using IP interconnectivity (IPIC). In this scenario, a CICS IPIC *TCPIPService* (see below) has been defined in the target CICS region to listen for input TCP/IP request on port 1491.

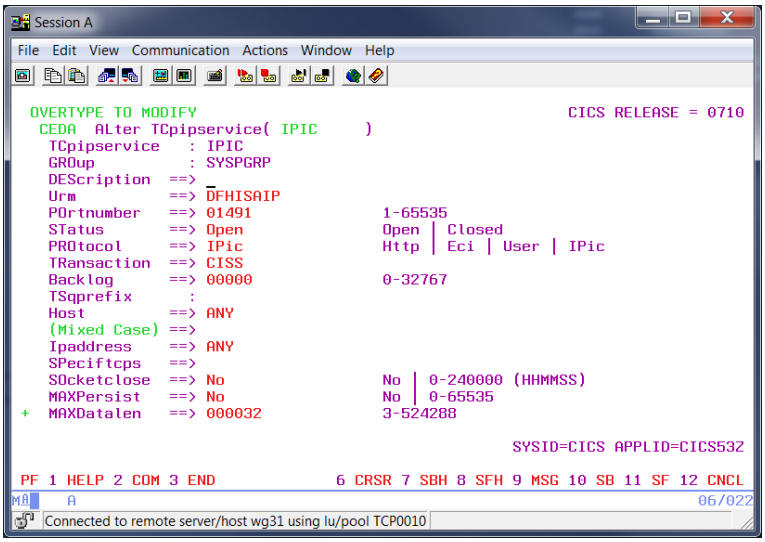
```
<server description="CICS IPIC - Catalog">

    <featureManager>
      <feature>zosconnect:cicsService-1.0</feature>
    </featureManager>

    <zosconnect_cicsIpicConnection id="catalog"
      host="wg31.washington.ibm.com"
      port="1491"/>

</server>
```

Figure 2: Contents of ipic.xml

Tech-Tip: Service names do not have to be explicitly identified in a *zosconnect_services* element. Services archive files in the *services* directory will be automatically installed either a server restart or based on the setting of the *updateTrigger* property (see *server.xml* above). One reason for adding explicit definitions for services (*zosconnect_services*) is when security attributes are required.

Mitch Johnson (mitchj@us.ibm.com)

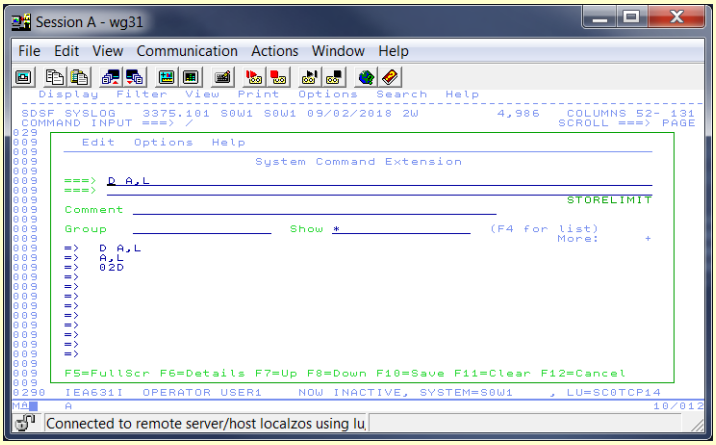
Note, the corresponding CICS TCPIPService definition.



___3. Use the **F3** key to end the edit session.

___4. Adding this file using an *include* statement refreshing the z/OS Connect EE V3.0 server. Enter the following MVS command to refresh the configuration ***F BAQSTRT,ZCON,REFRESH***

Tech-Tip: MVS commands can be only entered on any Spool Search and Display Facility (SDSF) ISPF panel if they are prefixed with a slash. To enter the MVS command ***D A,L*** then you must enter ***/D A,L*** at the Command ===> prompt on the panel. The best way to review the results of the command is to use the SDSF ***LOG*** command to access the system log and go to the time in the log the command was entered.
An alternative is to simply enter the SDSF / (slash) command at the command prompt to display the System Command Extension panel (see below).



This panel provides more space for enter command parameter and also provide a means to retrieve and execute previous command. Simply place the cursor on the command in the command history and press enter. This will retrieve the command and place in the active area where a subsequent pressing of the enter key will cause the command to be executed.

Mitch Johnson (mitchj@us.ibm.com)

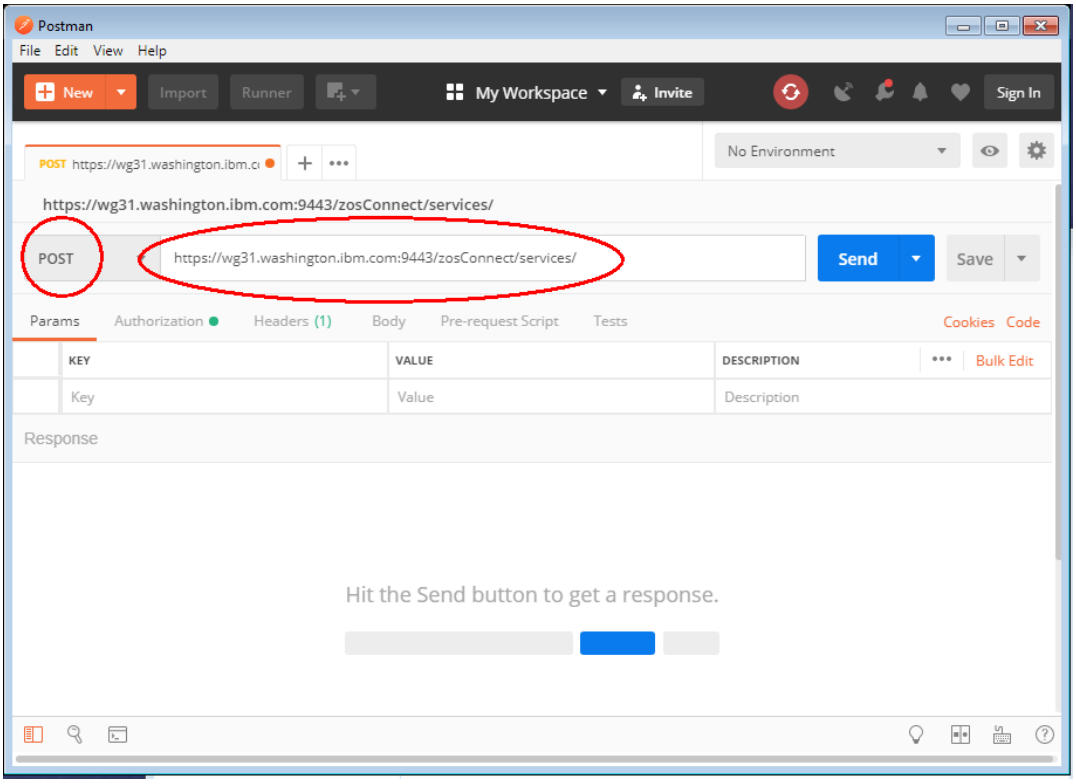
Deploy the Services

The common application artifacts installed into z/OS connect server are services and APIs (to see how these artifacts are developed, review any of the other exercises in this workshop). A *service* describes a specific interaction with a back end subsystem wheras an API describes the RESTful interface to one or more services. Both of these artifacts are archive (e.g. zipped) files. They should be installed using the z/OS Connect EE RESTful administrative interface. Two products which seem to be most popular tools for invoking z/OS Connect EE administrative RESTful APIs are *Postman* which is available for downloading from https://www.getpostman.com/apps and *cURL (client URL)* which is available for downloading from https://curl.haxx.se/download.html. The use of both *Postman* an *cURL* will be shown in this section of the exercise.

These instructions provide details on both. Choose either *Postman* or *cURL* to deploy the service.

Using Postman

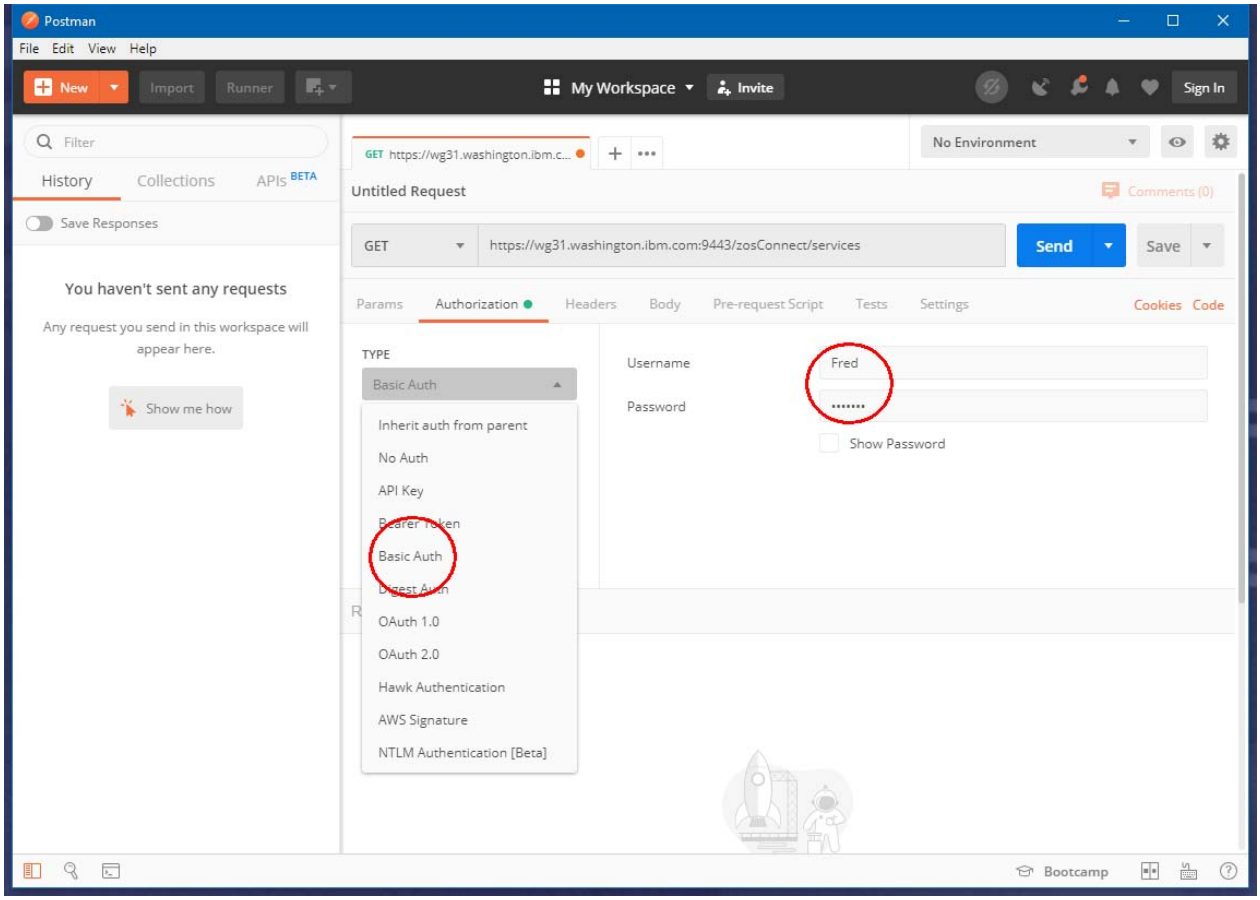
___1. Open the *Postman* tool icon on the desktop. If necessary reply to any prompts and close any welcome messages, use the down arrow to select **POST** and enter ***https://wg31.washington.ibm.com:9443/zosConnect/services/*** in the URL area (see below).



Tech-Tip: If the above Postman view is not displayed select *File* on the toolbar and then choose *New Tab* on the pull down. Alternatively, if the *Launchpad* view is displayed, click on the *Create a request* option.

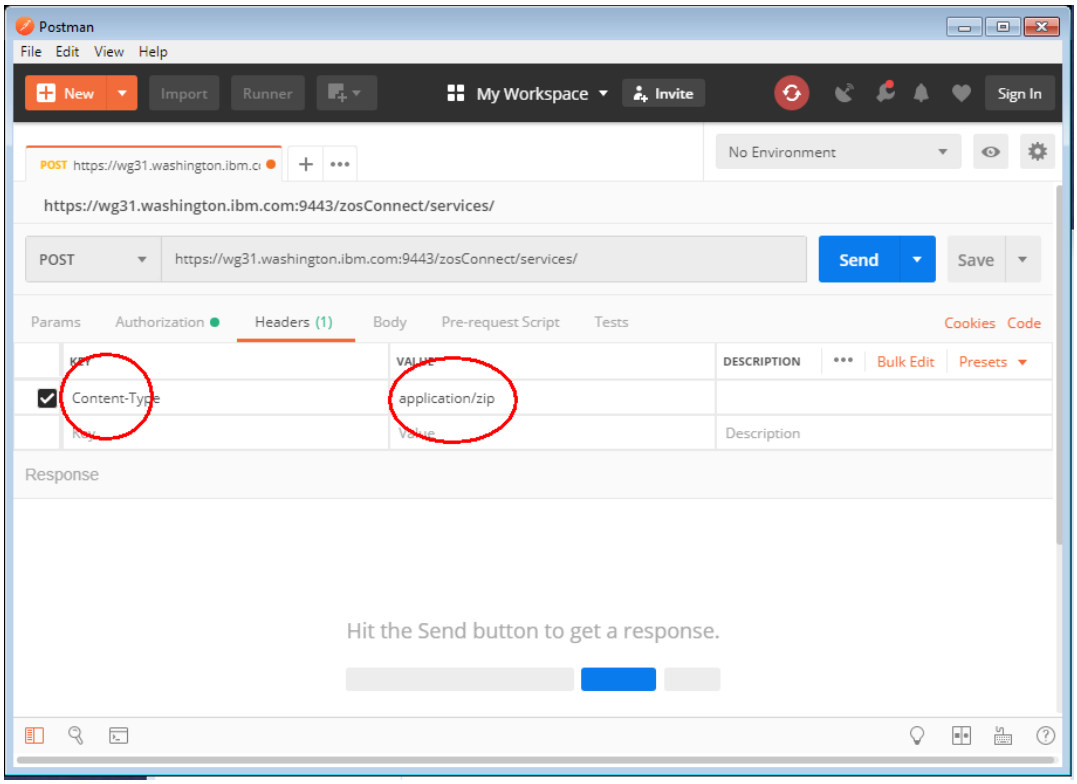
Mitch Johnson (mitchj@us.ibm.com)

___2. Next, select the *Authorization* tab to enter an authorization identity and password. Use the pull down arrow to select *Basic Auth* and enter ***Fred*** as the username and ***fredpwd*** as the Password.



Mitch Johnson (mitchj@us.ibm.com)

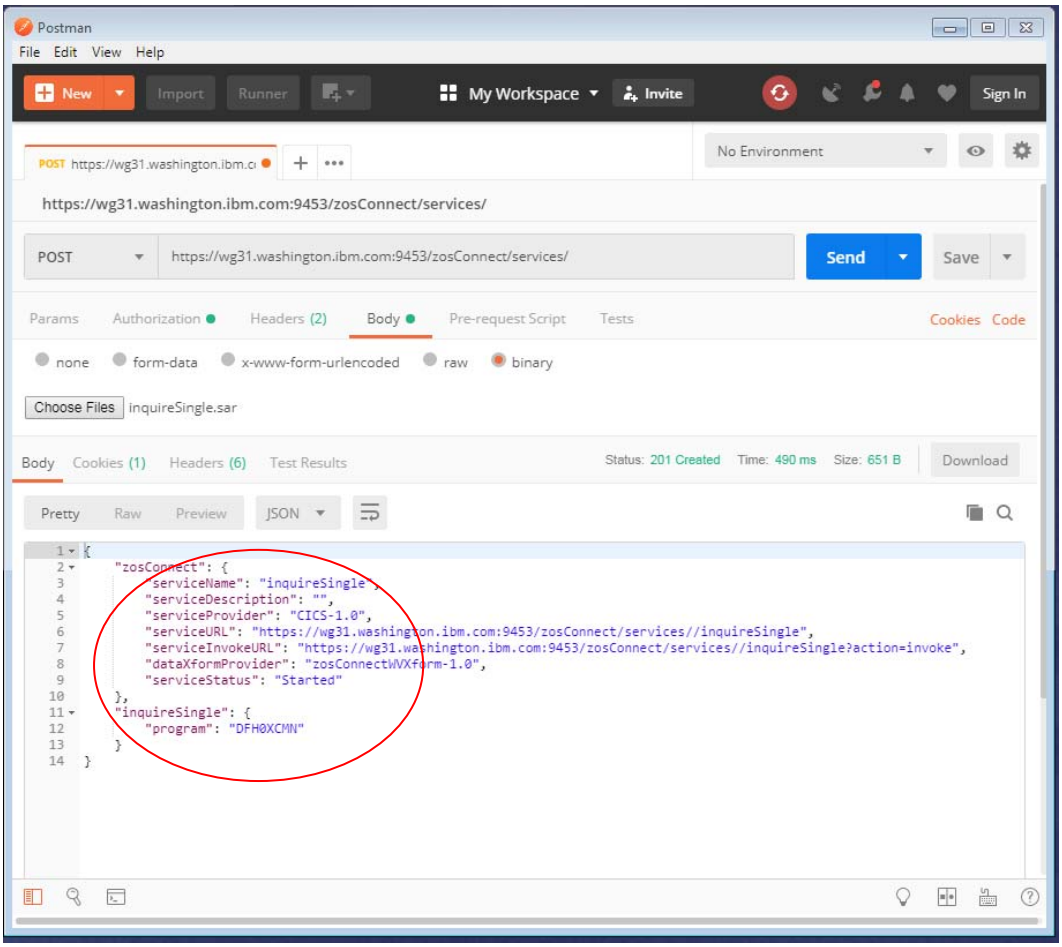
___3. Next, select the *Headers* tab. Under *KEY* use the code assist feature to enter ***Content-Type***, and under *VALUE*, use the code assist feature to enter ***application/zip***.



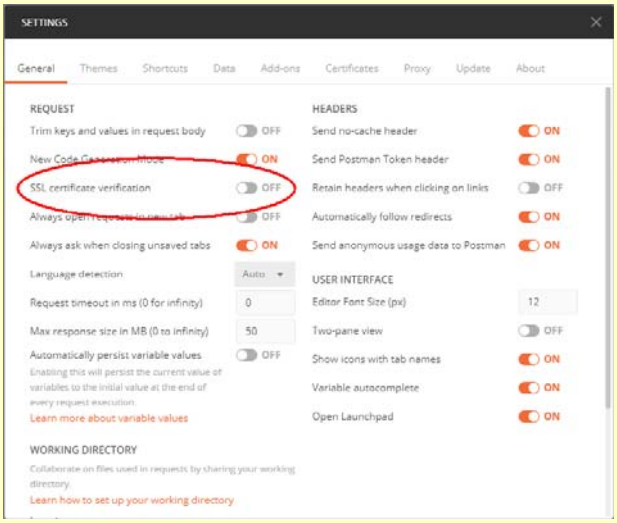
Tech-Tip: Code assist simply means that when text is entered in field, all the valid values for that field that match the typed text will be displayed. You can select the desired value for the field from the list displayed and that value will populate that field.

Mitch Johnson (mitchj@us.ibm.com)

___4. Next select the *Body* tab and select the *binary* radio button. Then use the **Choose Files** button to navigate to folder *c:/z/*admin. Select the *inquireSingle.Sar* file. Then press the **Send** button. A response message should come back indicating the service has been started and other details about the service.



Tech-Tip: The Postman settings have been changed to disable *SSL certificate verification*, a settings option.



Mitch Johnson (mitchj@us.ibm.com)

Using cURL

The *cURL* tool provides a command line interface to REST APIs. The same administrative API service invoked with *Postman* can be invoked with *cURL* as shown here.

___1. Use the *Command Prompt* icon on the desktop to open a DOS command prompt session.

___2. In the session use the change directory (cd) command to go to directory c:\z\admin, e.g.
cd c:\z\admin

___3. Paste the command below at the command prompt and press **Enter**.

curl -X POST --user Fred:fredpwd --data-binary @inquireSingle.sar --header "Content-Type: application/zip" --insecure https://wg31.washington.ibm.com:9443/zosConnect/services

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\workstation>cd c:\z\admin

c:\z\admin>curl -X POST --user Fred:fredpwd --header "Content-Type: application/zip"
-d @inquireSingle.json --insecure https://wg31.washington.ibm.com:9443/zosConnect/services
{"zosConnect":{"serviceName":"inquireSingle","serviceDescription":"",
"serviceProvider":"CICS-
1.0","serviceURL":"https://wg31.washington.ibm.com:9443/zosConnect/services/inquireSingle",
"serviceInvokeURL":"https://wg31.washington.ibm.com:9443/zosConnect/services/inquireSingle?action
=invoke",
"dataXformProvider":"zosConnectWVXform-1.0","serviceStatus":"Started"},
"inquireSingle":{"program":"DFH0XCMN"}}
```

Tech-Tip: In the above example:

--user Fred:fredpwd could have been specified as ***--header "Authorization: Basic RnJlZDpmcmVkcHdk"***

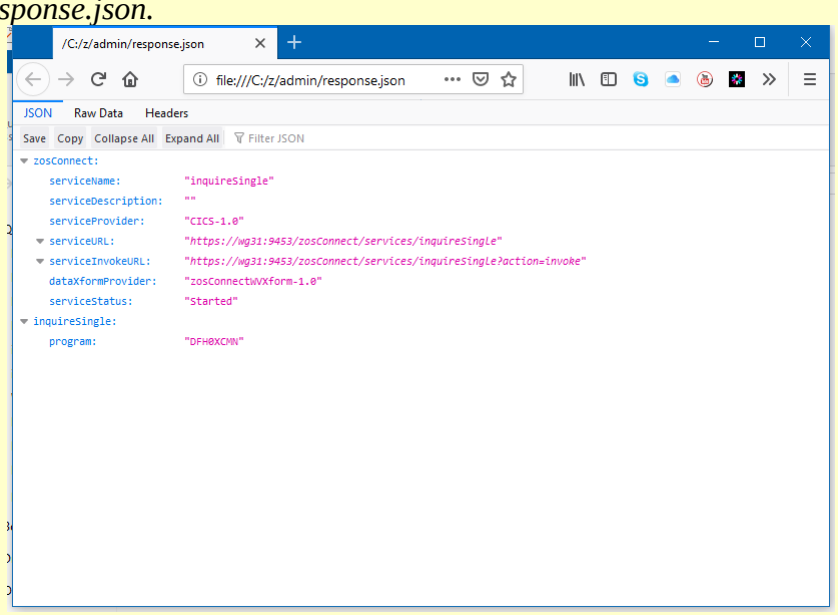
inquireCatalog.sar is a file in the same directory in which the command is executed.

--insecure is a *cURL* directive that tells *cURL* to ignore the self-signed certificate sent by the z/OS Connect EE server.

The text in green is the JSON response message.

Mitch Johnson (mitchj@us.ibm.com)

Tech-Tip: Another useful cURL directive is *-o response.json.*
When this directive is used, the JSON response message is written to a file named *response.json* which then can be opened with Firefox and viewed in a more readable format, e.g. command *firefox file:///c:/z/admin/response.json.*



___4. Use either *Postman* or *cURL* to deploy the *inquireCatalog.sar* file.

curl -X POST --user Fred:fredpwd --data-binary @inquireCatalog.sar --header "Content-Type: application/zip" --insecure https://wg31.washington.ibm.com:9443/zosConnect/services

___5. Use either *Postman* or *cURL* to deploy the *placeOrder.sar* file.

curl -X POST --user Fred:fredpwd --data-binary @placeOrder.sar --header "Content-Type: application/zip" --insecure https://wg31.washington.ibm.com:9443/zosConnect/services

Tech-Tip: The only differences between the commands is the name of the service archive file being deployed.

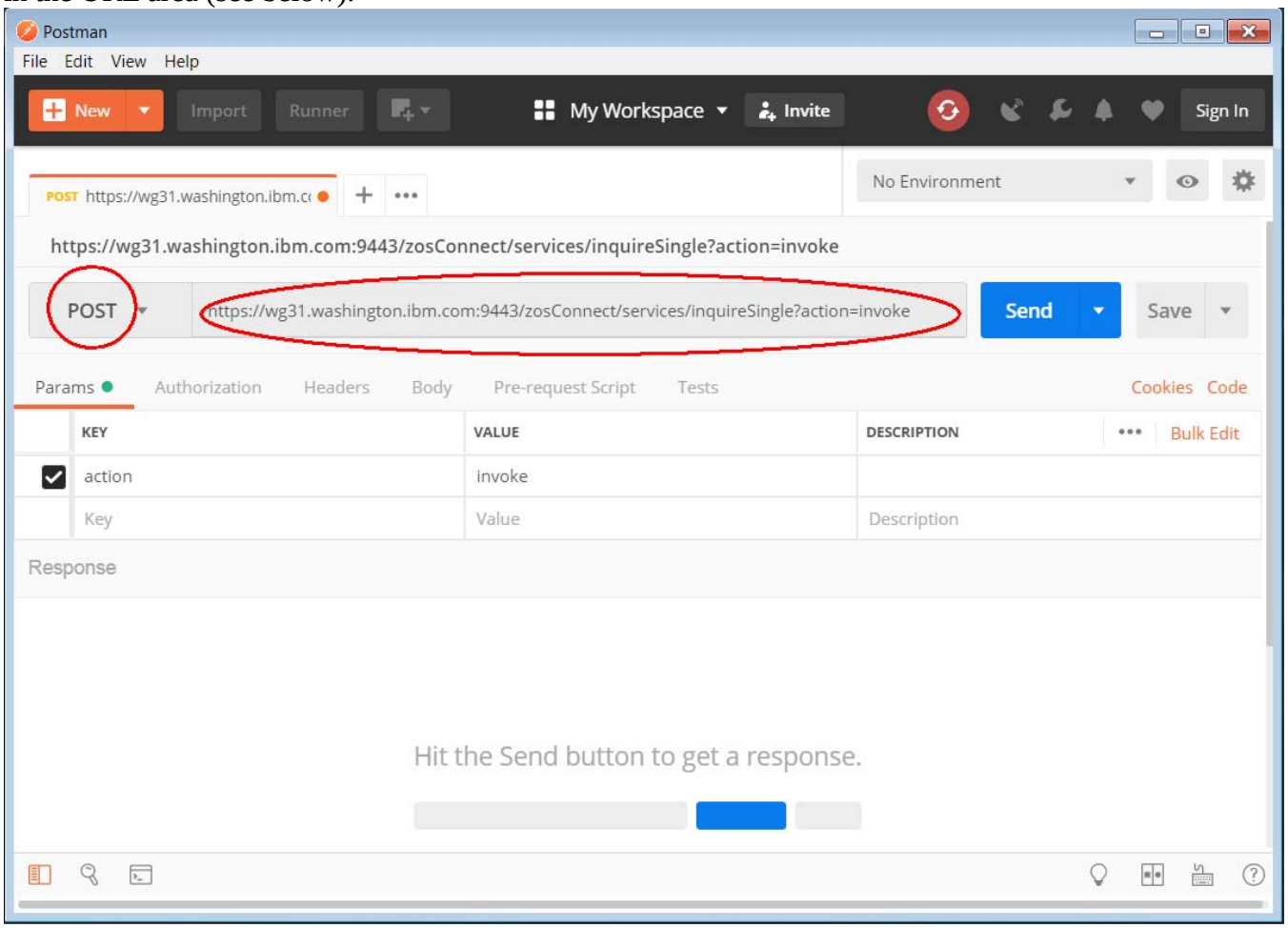
Mitch Johnson (mitchj@us.ibm.com)

Test the Services

The services should be tested to ensure the infrastructure and the request and response messages are as expected.

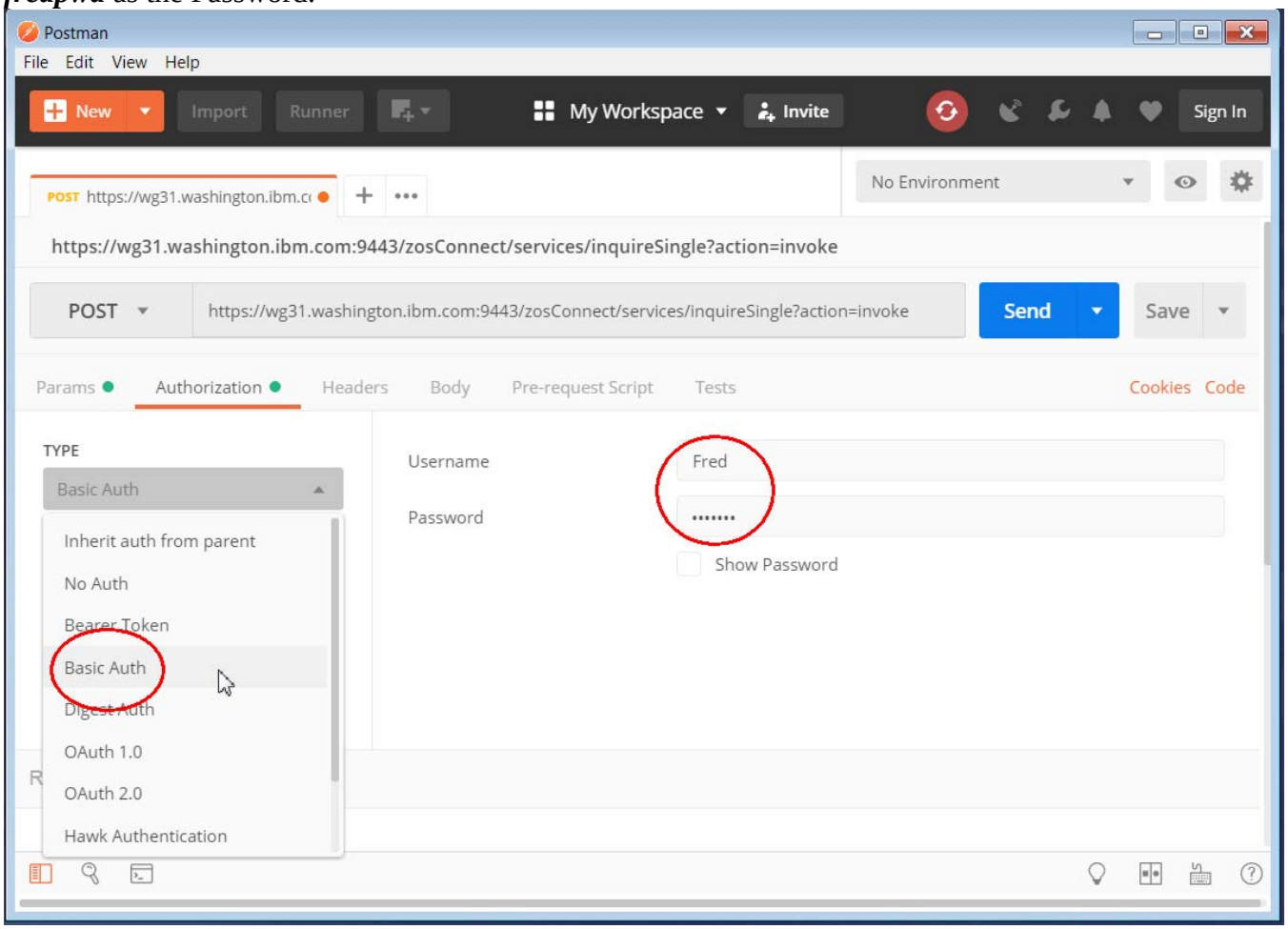
Using Postman

___1. Open the *Postman* tool icon on the desktop and if necessary, reply to any prompts. Close any welcome messages. The use the down arrow to select **POST** and enter ***https://wg31.washington.ibm.com:9443/zosConnect/services/inquireSingle?action=invoke*** in the URL area (see below).



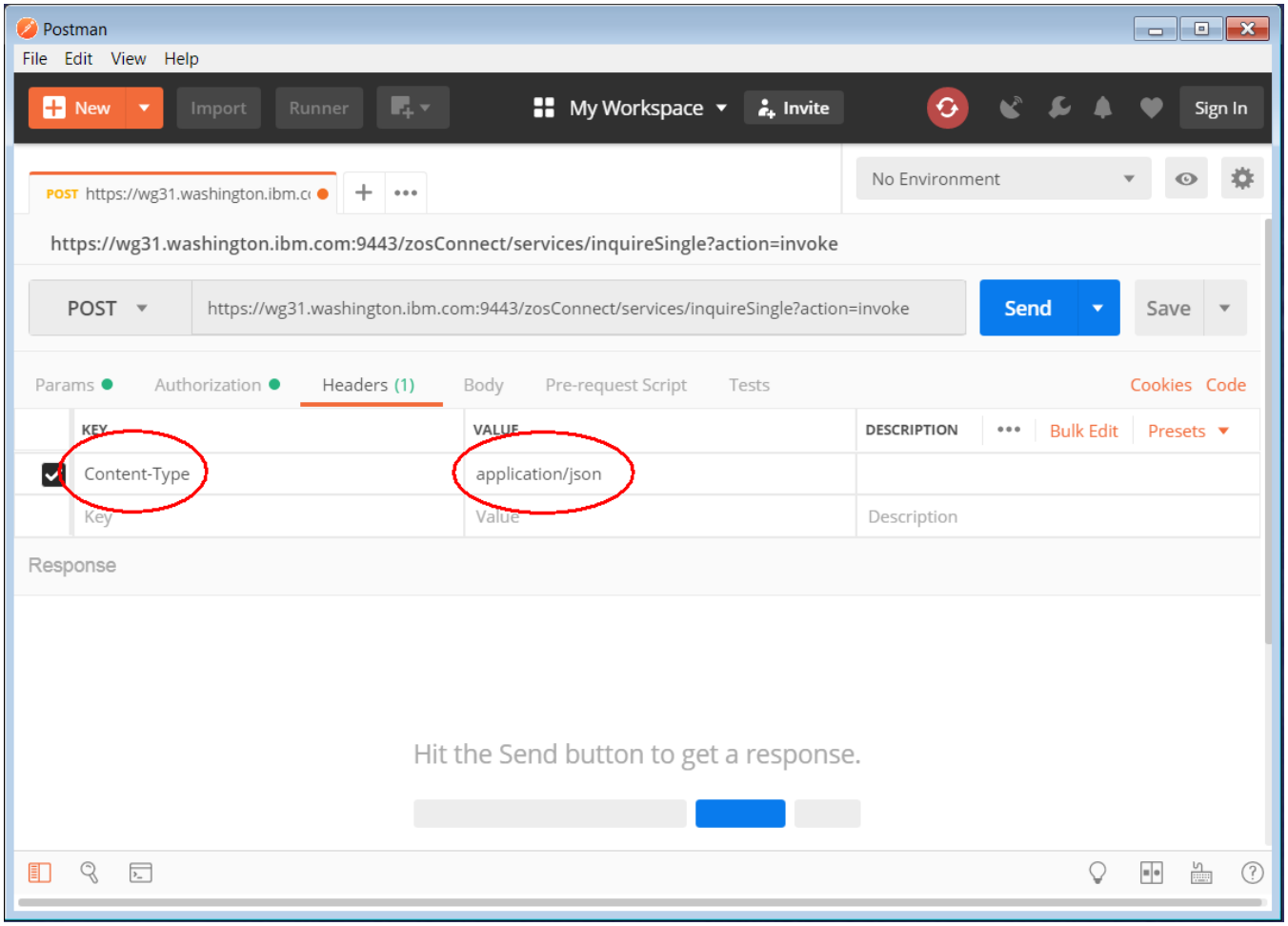
Mitch Johnson (mitchj@us.ibm.com)

___2. No *query* or *path* parameters are required so next select the *Authorization* tab to enter an authorization identity and password. Use the pull down arrow to select *Basic Auth* and enter ***Fred*** as the username and ***fredpwd*** as the Password.



Mitch Johnson (mitchj@us.ibm.com)

___3. Next, select the *Headers* tab. Under *KEY,* use the code assist feature to enter ***Content-Type,*** and under *VALUE,* use the code assist feature to enter ***application/json***.

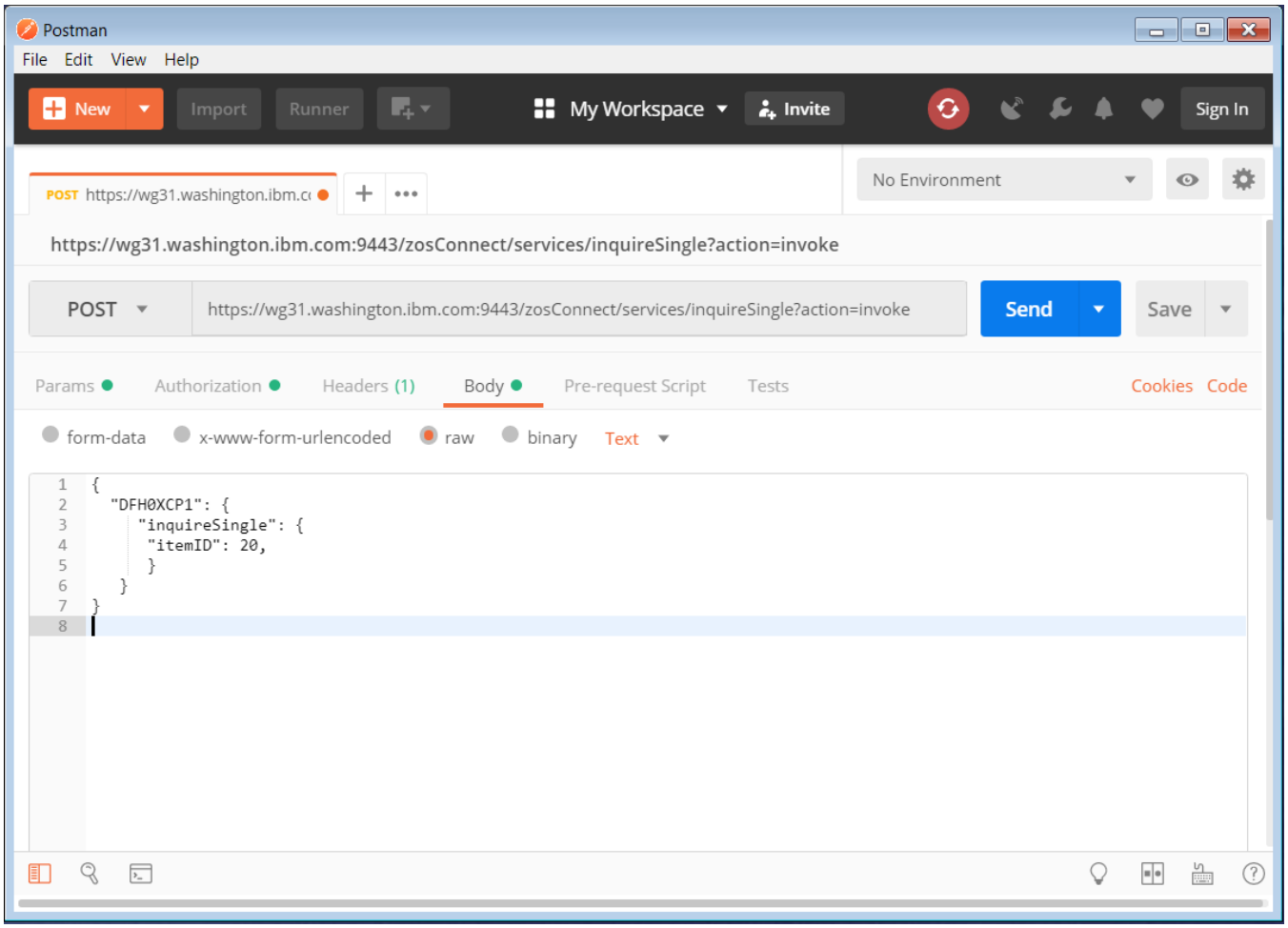


Tech-Tip: Code assist simply means that when text is entered in field, all the valid values for that field that match the typed text will be displayed. You can select the desired value for the field from the list displayed and that value will populate that field.

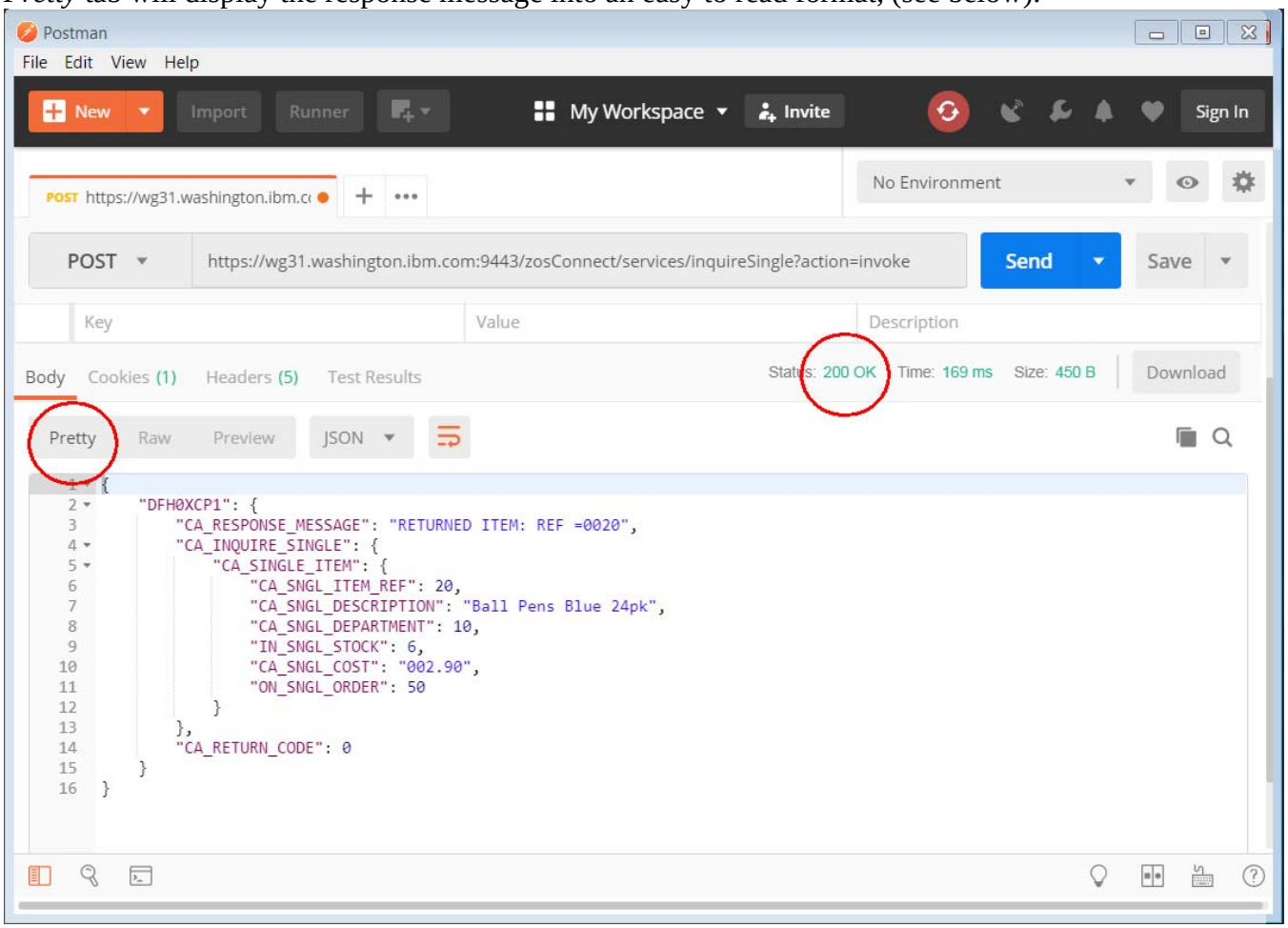
Mitch Johnson (mitchj@us.ibm.com)

___4. Next select the *Body* tab and select the *raw* radio button and enter the JSON message below in the *Body* area and press the S**end** button.

```
{
  "DFH0XCP1": {
     "inquireSingle": {
      "itemID": 20,
      }
   }
}
```



___5. Pressing the **Send** button invokes the API. The Status of request should be 200 OK, and pressing the *Pretty* tab will display the response message into an easy to read format, (see below).



Using cURL

The *cURL* tool provides a command line interface to REST APIs. The same service just tested with *Postman* can be tested with *cURL* as shown here.

___1. Use the *Command Prompt* icon on the desktop to open a DOS command prompt session.

___2. In the session use the change directory (cd) command to go to directory c:\z\admin, e.g.
cd c:\z\admin

___3. Paste the command below at the command prompt and press **Enter**.

curl -X POST --user Fred:fredpwd --header "Content-Type: application/json" -d @inquireSingle.json --insecure https://wg31.washington.ibm.com:9443/zosConnect/services/inquireSingle?action=invoke

jMicrosoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\workstation>***cd c:\z\admin***

c:\z\admin>***curl -X POST --user Fred:fredpwd --header "Content-Type: application/json" -d @inquireSingle.json --insecure https://wg31.washington.ibm.com:9443/zosConnect/services/inquireSingle?action=invoke***
{"DFH0XCP1":{"CA_RESPONSE_MESSAGE":"RETURNED ITEM: REF =0020","CA_INQUIRE_SINGLE":{"CA_SINGLE_ITEM":{"CA_SNGL_ITEM_REF":20,"CA_SNGL_DESCRIPTION":"Ball Pens Blue 24pk", "CA_SNGL_DEPARTMENT":10, "IN_SNGL_STOCK":6, "CA_SNGL_COST":"002.90", "ON_SNGL_ORDER":50}}, "CA_RETURN_CODE":0}}

Tech Tip: In the above example:

--user Fred:fredpwd could have been specified as ***--header "Authorization: Basic RnJlZDpmcmVkcHdk"***

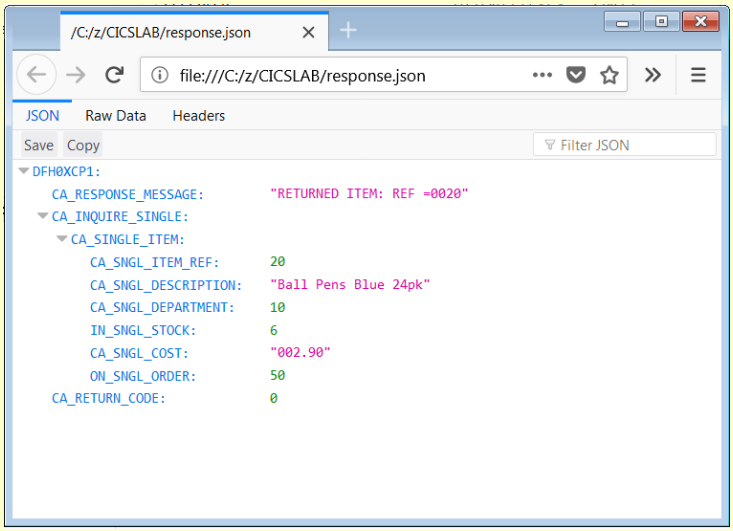
inquireSingle.json is a file in the same directory that contains the JSON request message.

--insecure is a *cURL* directive that tells *cURL* to ignore the self-signed certificate sent by the z/OS Connect EE server.

The text in green is the JSON response message.

Mitch Johnson (mitchj@us.ibm.com)

Tech-Tip: Another useful cURL directive is *-o response.json.*
When this directive is used, the JSON response message is written to a file named *response.json* which then can be opened with Firefox and viewed in a more readable format, e.g. command *firefox file:///c:/z/admin/response.json*



___4. The *inquireCatalog* service can be tested with *Postman* or *cURL* with URL
https://wg31.washington.ibm.com:9443/zosConnect/services/inquireCatalog?action=invoke
and JSON request message.

```
{
  "DFH0XCP1": {
     "inquireCatalog": {
      "startItemID": 20
      }
   }
}
```

The corresponding cURL command and results are shown below:

curl -X POST --user Fred:fredpwd --header "Content-Type: application/json"
-d @inquireCatalog.json --insecure
https://wg31.washington.ibm.com:9443/zosConnect/services/inquireCatalog?action=invoke

Mitch Johnson (mitchj@us.ibm.com)

```
c:\z\admin>curl -X POST --user Fred:fredpwd --header "Content-Type: application/json"
-d @inquireCatalog.json --insecure https://wg31.washington.ibm.com:9443/
zosConnect/services/inquireCatalog?action=invoke
{"DFH0XCP1":{"CA_RESPONSE_MESSAGE":"+15 ITEMS RETURNED","CA_INQUIRE_REQUEST":{"C
A_LAST_ITEM_REF":150,"CA_CAT_ITEM":[{"ON_ORDER":0,"CA_ITEM_REF":10,"CA_COST":"00
2.90","IN_STOCK":135,"CA_DESCRIPTION":"Ball Pens Black 24pk","CA_DEPARTMENT":10}
,{"ON_ORDER":50,"CA_ITEM_REF":20,"CA_COST":"002.90","IN_STOCK":6,"CA_DESCRIPTION
":"Ball Pens Blue 24pk","CA_DEPARTMENT":10},{"ON_ORDER":0,"CA_ITEM_REF":30,"CA_C
OST":"002.90","IN_STOCK":106,"CA_DESCRIPTION":"Ball Pens Red 24pk","CA_DEPARTMEN
T":10},{"ON_ORDER":0,"CA_ITEM_REF":40,"CA_COST":"002.90","IN_STOCK":80,"CA_DESCR
IPTION":"Ball Pens Green 24pk","CA_DEPARTMENT":10},{"ON_ORDER":0,"CA_ITEM_REF":5
0,"CA_COST":"001.78","IN_STOCK":83,"CA_DESCRIPTION":"Pencil with eraser 12pk","C
A_DEPARTMENT":10},{"ON_ORDER":40,"CA_ITEM_REF":60,"CA_COST":"003.89","IN_STOCK":
13,"CA_DESCRIPTION":"Highlighters Assorted 5pk","CA_DEPARTMENT":10},{"ON_ORDER":
20,"CA_ITEM_REF":70,"CA_COST":"007.44","IN_STOCK":101,"CA_DESCRIPTION":"Laser Pa
per 28-lb 108 Bright 500\/ream","CA_DEPARTMENT":10},{"ON_ORDER":0,"CA_ITEM_REF":
80,"CA_COST":"033.54","IN_STOCK":25,"CA_DESCRIPTION":"Laser Paper 28-lb 108 Brig
```

The *placeOrder* service can be tested using *Postman* or *cURL* with URL:
https://wg31.washington.ibm.com:9443/zosConnect/services/placeOrder?action=invoke

and JSON request message.

```
{
  "DFH0XCP1": {
     "orderRequest": {
      "itemID": 70,
      "orderQuantity": 1
      }
   }
}
```

The corresponding cURL command and results are shown below:

curl -X POST --user Fred:fredpwd --header "Content-Type: application/json" -d @placeOrder.json --insecure https://wg31.washington.ibm.com:9443/zosConnect/services/placeOrder?action=invoke

```
c:\z\admin>curl -X POST --user Fred:fredpwd --header "Content-Type: application/json"
 -d @placeOrder.json --insecure https://wg31.washington.ibm.com:9443/zosC
onnect/services/placeOrder?action=invoke
{"DFH0XCP1":{"CA_RESPONSE_MESSAGE":"ORDER SUCCESSFULLY PLACED","CA_RETURN_CODE":
0}}
```


Mitch Johnson (mitchj@us.ibm.com)

Deploy and Test the API

The API artifact should be deployed using z/OS Connect RESTful administrative interface. cURL will be used in this section, but Postman could have been used also.

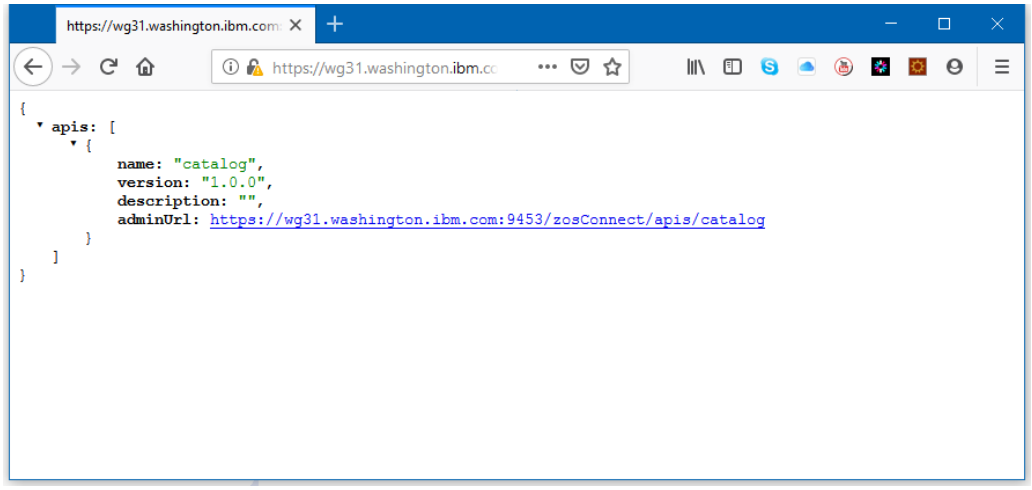
___1. Deploy the API archive file using cURL. Paste the command below at the command prompt and press **Enter**.

curl -X POST --user Fred:fredpwd --data-binary @catalog.aar --header "Content-Type: application/zip" --insecure https://wg31.washington.ibm.com:9443/zosConnect/apis

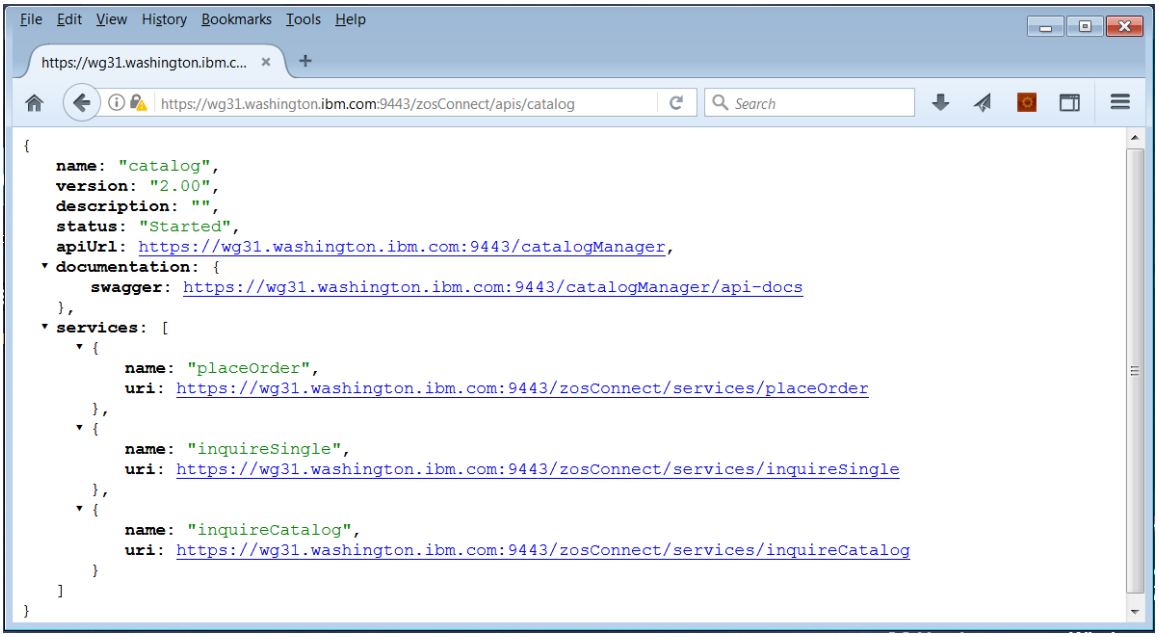
```
C:\z\admin>curl -X POST --user Fred:fredpwd --data-binary @catalog.aar --header
"Content-Type: application/zip"  --insecure
https://wg31.washington.ibm.com:9443/zosConnect/apis
{"name":"catalog","version":"1.0.0","description":"","status":"Started",
"apiUrl":"https://wg31.washington.ibm.com:9453/catalog",
"documentation":{"swagger":"https://wg31.washington.ibm.com:9443/catalog/api-docs"},
"services":[
{"name":"placeOrder","uri":"https://wg31.washington.ibm.com:9443/zosConnect/services/place
Order"},
{"name":"inquireSingle","uri":"https://wg31.washington.ibm.com:9443/zosConnect/services/inq
uireSingle"},
{"name":"inquireCatalog","uri":"https://wg31.washington.ibm.com:9443/zosConnect/services/in
quireCatalog"}
]}
```


Mitch Johnson (mitchj@us.ibm.com)

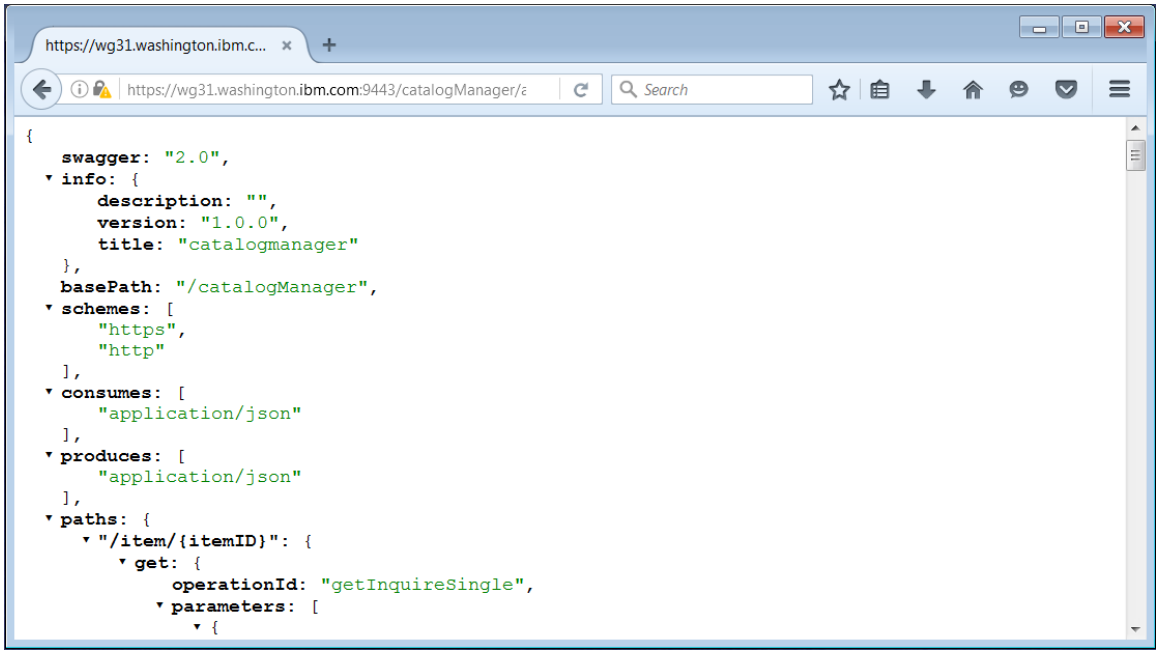
___2. Next enter URL ***https://wg31.washington.ibm.com:9443/zosConnect/apis*** in the Firefox browser and you should see the window below. The *catalog* API now shows as being available.



___3. If you click on *adminUrl* URL, the window below should be displayed.

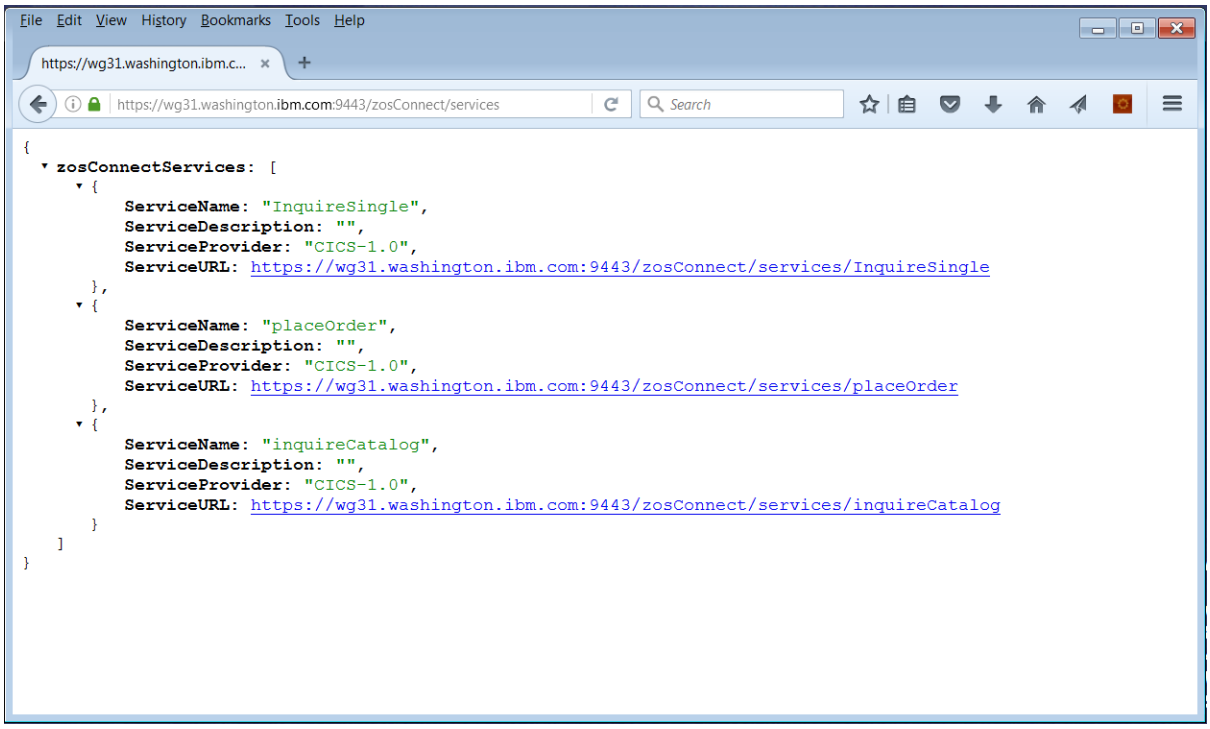


___4. Next click on the *swagger* URL and you should see the Swagger document associated with this API:



Explore this Swagger document and you will see the results of the request and response mapping performed earlier. This Swagger document can be used by a developer or other tooling to develop REST clients for this specific API.

___5. Next enter URL ***https://wg31.washington.ibm.com:9443/zosConnect/services*** in the Firefox browser and you should see the window below. The three services are now available.



Mitch Johnson (mitchj@us.ibm.com)

Use cURL or Postman to test the API.

curl -X GET --user Fred:fredpwd --insecure https://wg31.washington.ibm.com:9443/catalog/item/0010

curl -X GET --user Fred:fredpwd --insecure https://wg31.washington.ibm.com:9443/catalog/items?startItemID=0010

curl -X POST --user Fred:fredpwd --data @placeOrder.json --header "Content-Type: application/json" --insecure https://wg31.washington.ibm.com:9443/catalog/order

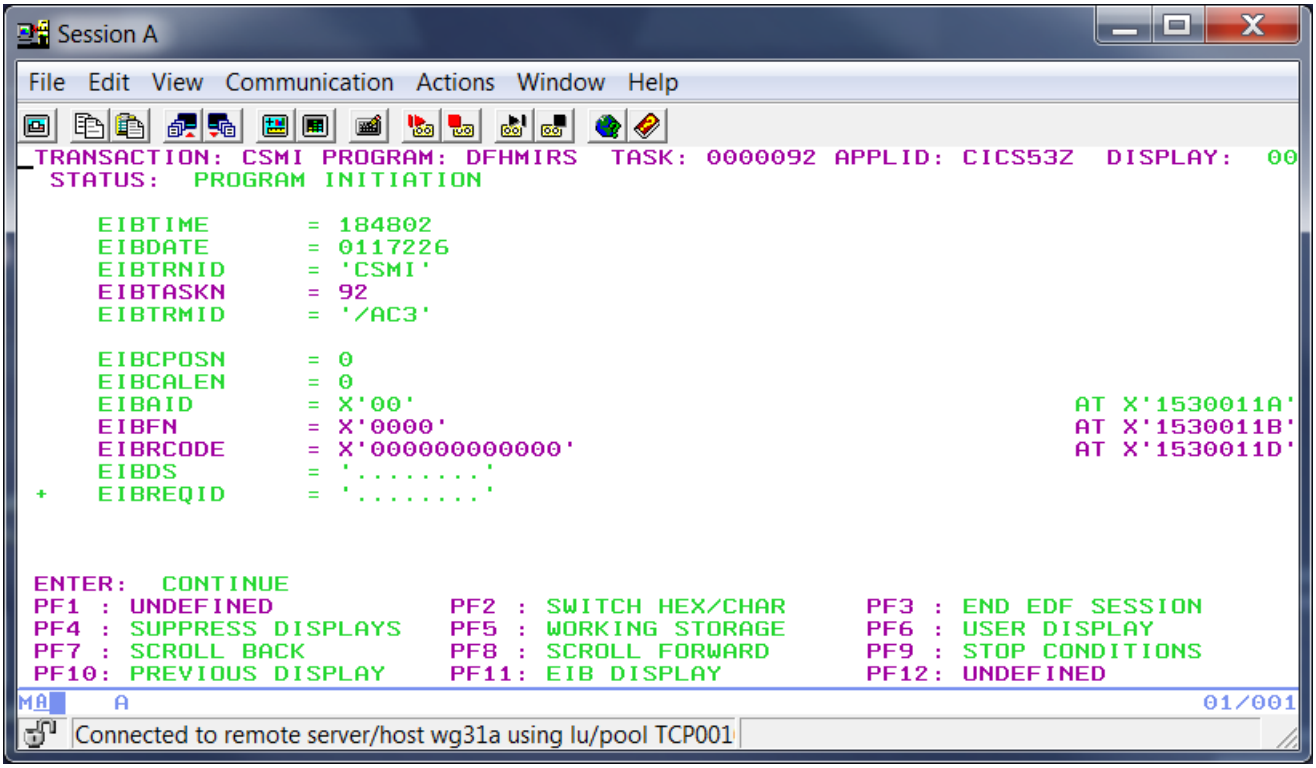
The available catalog items are listed below.

Item#	Description	Dept	Cost	In Stock	On Order
0010	Ball Pens Black 24pk	010	002.90	0135	000
0020	Ball Pens Blue 24pk	010	002.90	0006	050
0030	Ball Pens Red 24pk	010	002.90	0106	000
0040	Ball Pens Green 24pk	010	002.90	0080	000
0050	Pencil with eraser 12pk	010	001.78	0083	000
0060	Highlighters Assorted 5pk	010	003.89	0013	040
0070	Laser Paper 28-lb 108 Bright 500/ream	010	007.44	0102	020
0080	Laser Paper 28-lb 108 Bright 2500/case	010	033.54	0025	000
0090	Blue Laser Paper 20lb 500/ream	010	005.35	0022	000
0100	Green Laser Paper 20lb 500/ream	010	005.35	0003	020
0110	IBM Network Printer 24 - Toner cart	010	169.56	0012	000
0120	Standard Diary: Week to view 8 1/4x5 3/4	010	025.99	0007	000
0130	Wall Planner: Eraseable 36x24	010	018.85	0003	000
0140	70 Sheet Hard Back wire bound notepad	010	005.89	0084	000
0150	Sticky Notes 3x3 Assorted Colors 5pk	010	005.35	0036	045
0160	Sticky Notes 3x3 Assorted Colors 10pk	010	009.75	0067	030
0170	Sticky Notes 3x6 Assorted Colors 5pk	010	007.55	0064	030
0180	Highlighters Yellow 5pk	010	003.49	0088	010
0190	Highlighters Blue 5pk	010	003.49	0076	020
0200	12 inch clear rule 5pk	010	002.12	0014	010
0210	Clear sticky tape 5pk	010	004.27	0073	000

Mitch Johnson (mitchj@us.ibm.com)

Optional

If you are familiar with CICS Execution Diagnostic Facility (EDF), start a 3270-terminal session with CICS. Clear the screen and enter CICS transaction ***CEDX CSMI***. When you repeat of any of the above test, you should be able to trace the flow of the request through CICS.



Summary

You have verified the API. The API layer operates above the service layer you defined. The API layer provides a further level of abstraction, allows a more flexible use of HTTP verbs, and provides better mapping of data via the API editor function.

Mitch Johnson (mitchj@us.ibm.com)