IBM – CICS TS Workshop

L01 – CICS – Resource Builder Lab

Lab Version V0.1

October, 2022

Please send any comments on this lab exercise to:
SangSoo HAN or Manjunath D
sshan@sg.ibm.com
manjud46@sg.ibm.com

Overview

IBM CICS® Transaction Server (TS) resource builder is a DevOps utility that provides a way to automate the creation of CICS resource definitions using a *configuration as code* approach.

CICS resource definitions are closely tied to the function of an application and frequently need tailoring for the application. But, at a system level, they have wide ranging impacts. They must be organized and controlled and the authority to change them is typically with a system programmer. CICS resource builder provides CICS application developers with a controlled way of creating and modifying CICS resource definitions. They can be confident that what they do meet application and system standards and is approved by CICS system programmers. CICS system programmers can be confident that application developers can only make changes to CICS resource definitions within a scope that they have set.

The configuration of application resources is defined in *YAML (YAML Ain't Markup Language*). This can be stored directly in source control management, such as Git, along with other application code for version control and automation. CICS resource builder fits into a DevOps pipeline to allow you to easily orchestrate and provision your CICS applications.

CICS resource builder has a command-line interface that can be run from Linux®, Linux on Z, macOS, Windows, or z/OS® UNIX System Services shell or JCL. CICS resource builder works with the CSD update batch utility, DFHCSDUP, in CICS Transaction Server for z/OS to update the resource definitions that are used in the CICS TS region.

As a result, developers have a quicker turnaround time to getting resources defined in the CICS region. System programmers capture their best practices and standards in configuration to automate the process in a controlled way.

For reference, the CICS resource builder can be downloaded from the IBM Support page here, and documentation is available here. The resource builder has already been installed for the lab.

Lab Scenario

1. Extract CSD definitions and import it to a YAML file.
2. Generate the DFHCSD commands from the YAML file and install the resources into another CICS region.

Lab Requirements

Please note that there are often several ways to perform functions in and for CICS. This lab exercise will present one of the ways. If you are familiar with CICS, you will notice that some of the statements are general, and not necessarily true for every situation.

This lab uses the PCOMM and CICS Explorer. If you are not familiar with these, please contact one of the lab instructors for assistance.

The following are other assumptions made in this lab exercise.

- **CICS TS V6.1:** This lab exercise only works in CICS V6.1. You have your own z/OS image you can change all resources in four CICS regions.

- **Login:** A TSO userid is available with the appropriate password provided, and you will also use the same TSO userid with the z/OS Explorer.

- The **CICS Explorer:** In the lab environment we have installed the CICS Explorer to configure CICS resources and the security request recording function.

Lab Step Overview

Part 1: Try the CBSA (CICS Bank Sample Application)

Part 2: Extract CSD definitions by using the DFHCSUP

Part 3: Import the CSD definitions and create the YAML definitions

Part 4: Create the CSD definitions into the new group

Part 5: Run the batch job to define the new group into the CICS61F1 region

Part 6: Restart the CICS region (CICS61F1) and then, test the CBSA application in the CICS61F1 region

Part 7: Summary

Part 0: Check the CICS Explorer Connection to CICS

In this part of the lab exercise you will configure the connection between the CICS Explorer running on your workstation to CICS running on z/OS.

Start the CICS Explorer

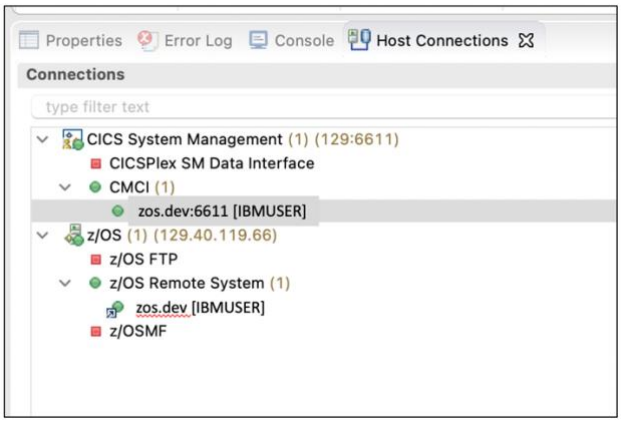
___1. From the **desktop**, **double-click** the **IBM Explorer for CICS and zOS** icon to start the Explorer if it is not already running.

___2. When you start the Explorer, if you are prompted for a workspace, click the **OK** button to select the default.

Verify that you have a FTP connection to z/OS and a CMCI connection to CICSPlex

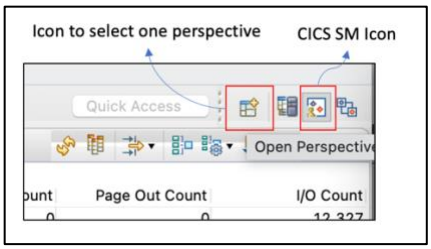
___3. If you have not already created connections to the z/OS host system, check the connection as in the screen shot. Both the **CMCI (6611) and z/OS Remote System** connections should be started and active.

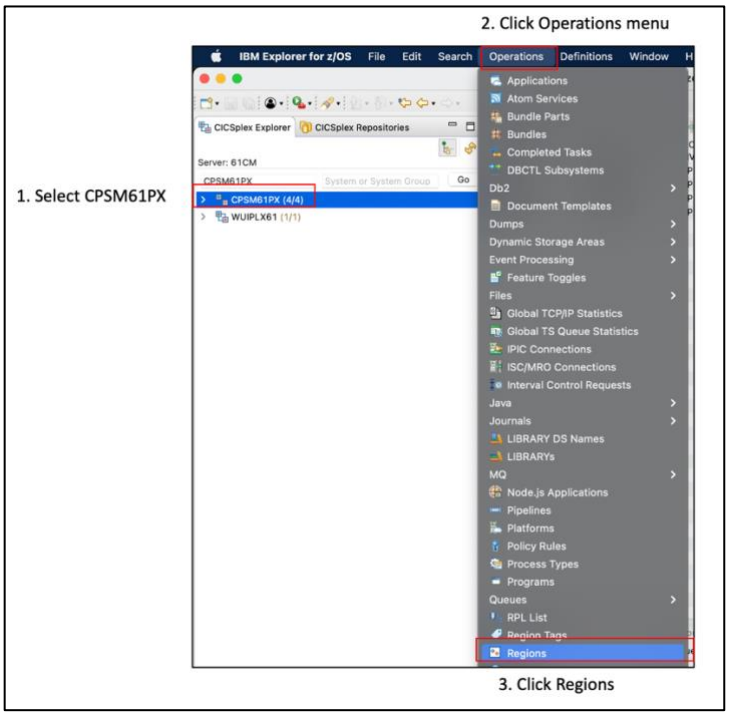
z/OS host name	`zos.dev`
z/OS userid	`IBMUSER`
z/OS user password	`sys1`
CICS CMCI port	`6611`



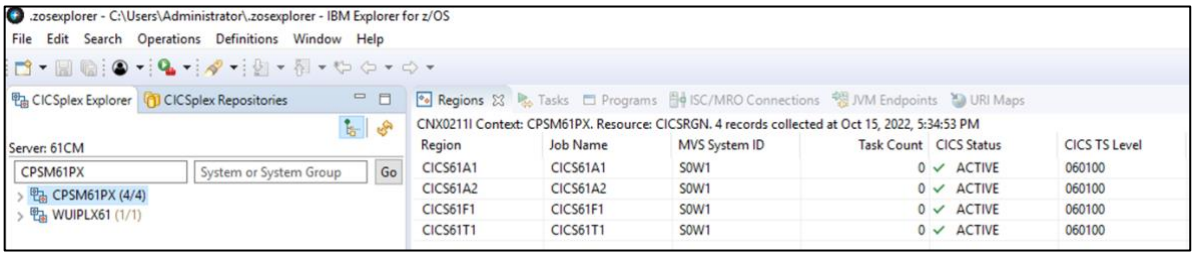
___4. Check the 4 CICS regions in the CICS explorer. Click “Operation” menu and region view.

At the top right corner:

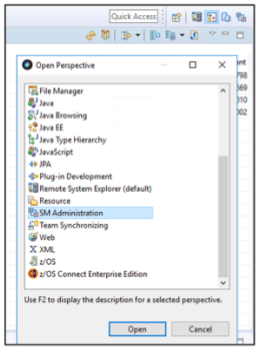




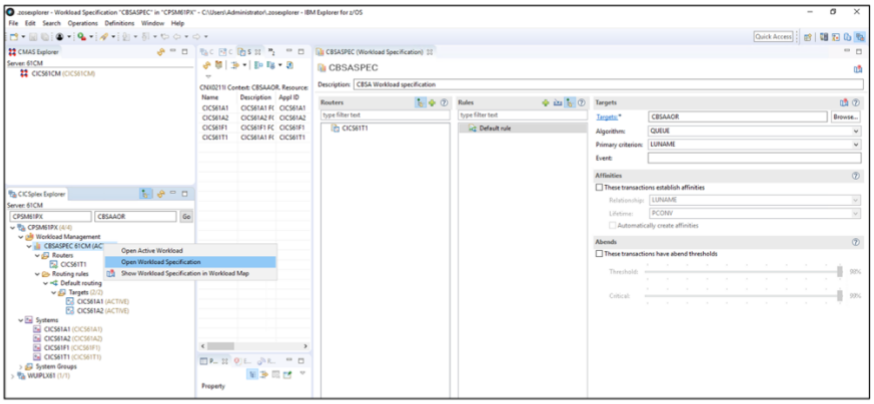
Check 4 CICS regions. CICS61T1, A1, A2 and F1



___5. Check the CICSPlex workload balancing definition as follows:



In the perspective menu,
Select “SM Administration”



Expand “Workload Management” in CICSPlex Explorer.
Select “CBSASPEC”, right click and select “Open Workload Specification”
Check the workload is active from CICS61T1 to CICS61A1 and CICS61A2.

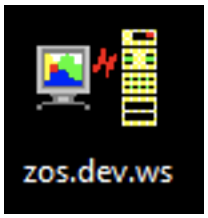
Just check the workload specification, CBSASPEC is active. No action is required.

Part 1: Try the CICS bank sample application (hereafter CBSA) in 4 CICS regions

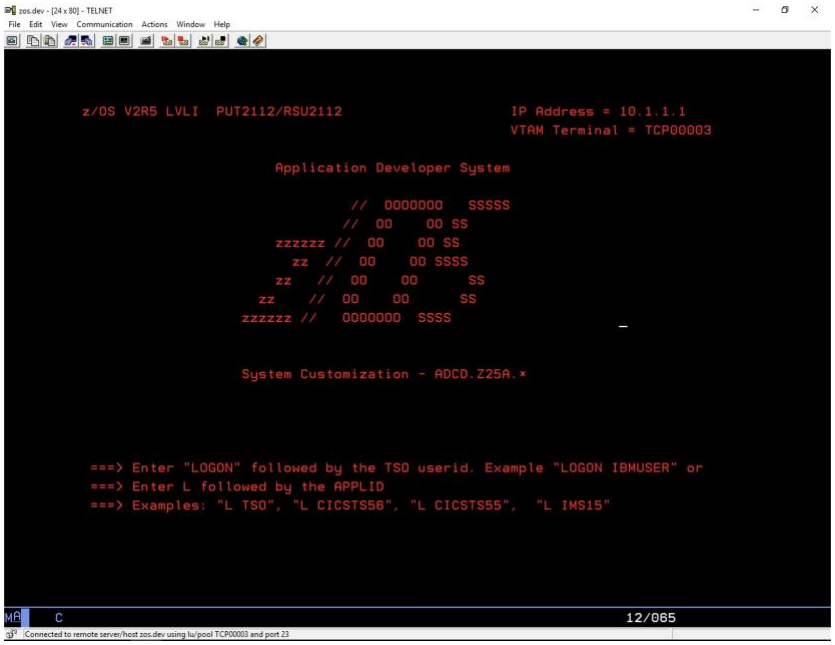
In this part of the lab exercise you will try the CICS bank sample application.

Try the CICS Bank Sample Application

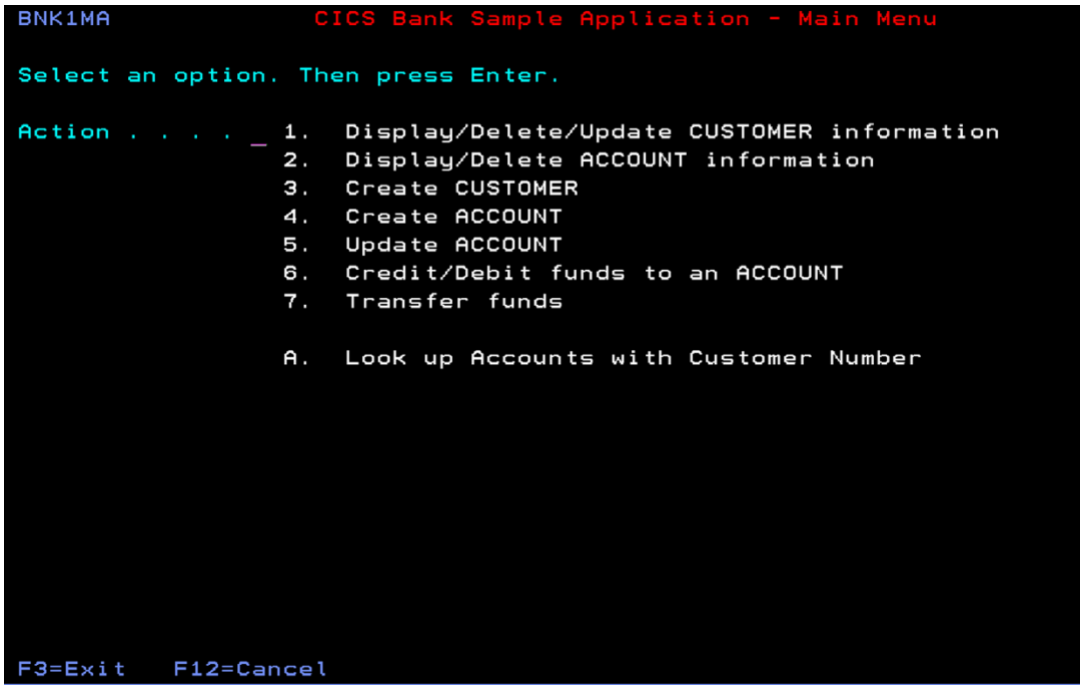
___1. At the Windows desktop, open a 3270 session using Personal Communications (PCOMM) by double-clicking on the "zos.dev.ws" icon.



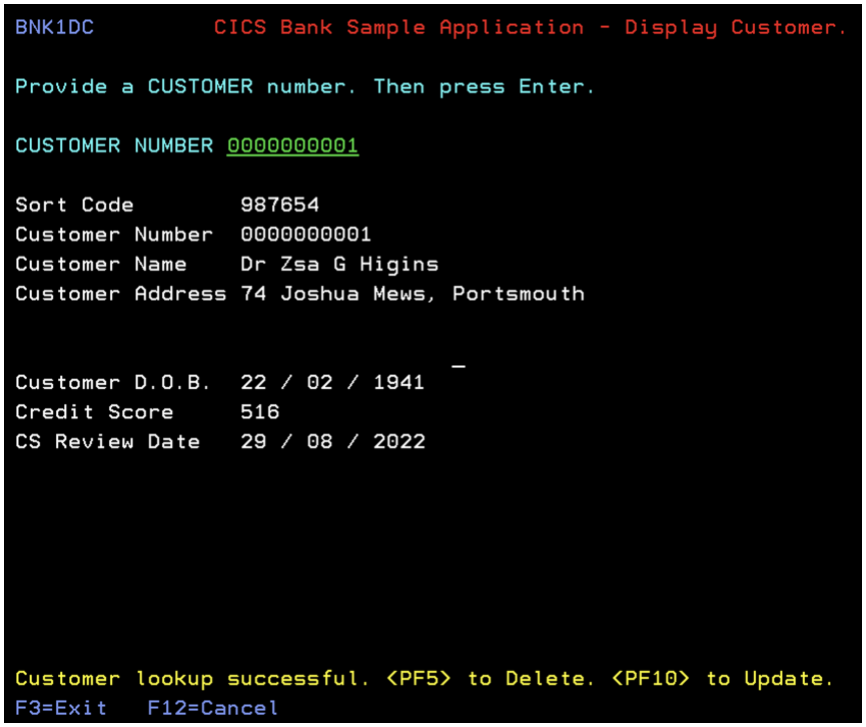
___2. At the initial 3270 panel, log on to a CICS region by typing "**`L CICS61T1`**" [ENTER].



___3. Clear the screen by [ESC] key and start the CICS Bank Sample Application by typing the CICS transaction ID "**OMEN**" [ENTER] to see the main menu.



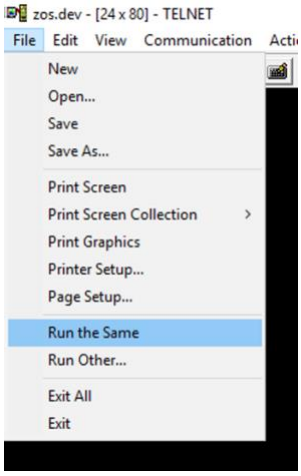
___4. Display the customer number 1 and the account number 1. No leading zeros are required to put.



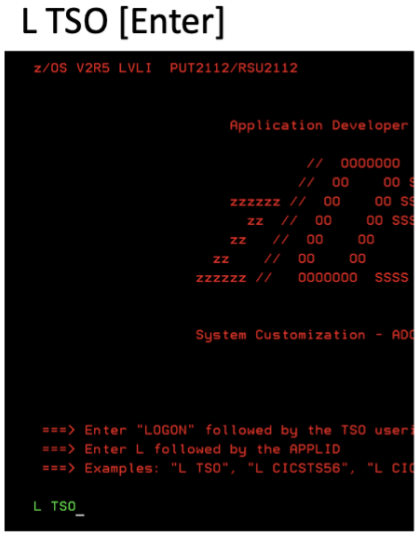
Part 2: Extract CSD definitions by using the DFHCSUP

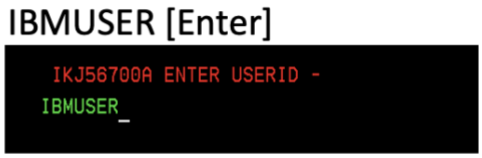
The resources for the application are in the CICS CSD file, within group BANK. We are going to extract all resource definitions from this group into a text file on z/OS, then transfer the text file to Windows.

1. Start a second PCOMM session by selecting “Run the Same” menu in “File” Menu in the current PCOMM session.



2. Log onto TSO by typing “L TSO” [ENTER]. The z/OS user id and password are shown above.





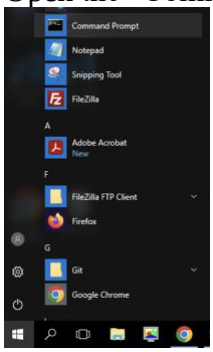
SYS1 [Enter]

3. Extract the resources from the CSD group named BANK. This can be done by submitting the following JCL which will copy the output to the CBDOUT dataset.

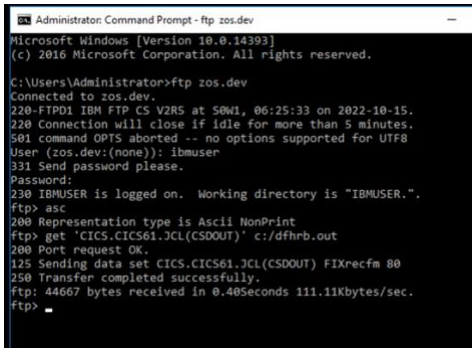
   =3.4
   move to “CICS.CICS61.JCL”
   submit the “DFHCSD”

4. FTP the extracted output to a USS file using ASCII mode.

   Open the “Command Prompt” app in the main windows menu.




```
ftp zos.dev
> ibmuser
> sys1
> asc
> get 'CICS.CICS61.JCL(CSDOUT)' c:/dfhrb.out
```



5. Replace the “GROUP(BANK)” to “GROUP(BANKRB)” in the file `c:/dfhrb.out`. It’s for making a new group in the same CSD for another CICS region, CICS61F1.

Part 3: Import the CSD definitions and create the YAML definitions

a. *Identify resources in CSD*

1. In the opened Windows command prompt, run the zrb import command, inputting the DFHCSDUP extracted file, dfhrb.out:

   Command:

   zrb import --source c:/dfhrb.out --import c:/cics-import.yaml --output-dir c:/bundleOutput/

   Within the command above, several arguments are used:

   - The --source argument is the path to the input file of resource definition commands, and the short name is -s.
   - The --import argument is the path to the import YAML file that defines the resource imports, and the short name is -i.
   - The --ouput-dir argument is the path to the output directory, and the short name is -o

- cics-import.yaml to setup parameters for the utility

```yaml
resourceImport:
  destinations:
    application: cics-bankrb-application.yaml
    model: cics-bankrb-model.yaml
    resources: cics-bankrb-resources.yaml
  applicationInput:
    name: My BankRB Application
    description: Import BankRB application tutorial
  modelInput:
    target: cicsts-6.1.0
  imports:
    - type: file
    - type: program
    - type: transaction
    - type: enqmodel
    - type: library
    - type: db2entry
    - type: db2tran
    - type: tcpipservice
    - type: pipeline
    - type: db2conn
```

➔ command execution result :

```
Totals:
  Imported:
    DB2CONN:                  1
    DB2ENTRY:                 1
    DB2TRAN:                 11
    ENQMODEL:                 1
    FILE:                     2
    LIBRARY:                  1
    PIPELINE:                 1
    PROGRAM:                 59
    TCPIPSERVICE:             2
    TRANSACTION:             18

Unselected resources:

Output directory files created:
  Resource model:            cics-bankrb-model.yaml
  Application constraints:   cics-bankrb-application.yaml
  Resource definitions:      cics-bankrb-resources.yaml
```

Part 4: Create the CSD definitions into the new group

CICS® resource builder uses the YAML files to validate and build a DFHCSDUP commands file, using the `zrb build` command. The output of zrb-build command is used as input to the CSD update utility, DFHCSDUP.

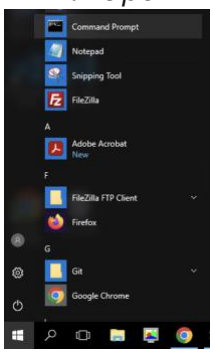
The following command generates a DFHCSDUP commands file (BankrbYaml.out) from a resource model that has an application constraints file, by using the optional `--application` argument of the `zrb build` command.

> *zrb build --model c:/bundleOutput/cics-bankrb-model.yaml --application c:/bundleOutput/cics-bankrb-application.yaml --resources c:/bundleOutput/cics-bankrb-resources.yaml --output c:/bundleOutput/BankrbYaml.out*
>
> ***➔ In the file, "C:\cicsresourcebuilder-Commands.txt", you can copy this command in the second line, and paste it to the command prompt window to run it.***

Part 5: Run the batch job to define the new group into the CICS61F1 region

Ftp the *BankrbYaml.out to z/OS dataset*

1. *Open the "Command Prompt" app in Windows*



2. *ftp commands to upload :*

ftp zos.dev
IBMUSER
Sys1
asc
put C:\bundleOutput\BankrbYaml.out 'CICS.CICS61.JCL(BANKRB)'

➔ Screen display from the above commands to compare your result:

```
C:\Users\Administrator>ftp zos.dev
Connected to zos.dev.
220-FTPD1 IBM FTP CS V2R5 at S0W1, 12:24:01 on 2022-10-10.
220 Connection will close if idle for more than 5 minutes.
501 command OPTS aborted -- no options supported for UTF8
User (zos.dev:(none)): IBMUSER
331 Send password please.
Password:
230 IBMUSER is logged on.  Working directory is "IBMUSER.".
ftp>
ftp> asc
200 Representation type is Ascii NonPrint
ftp> put C:\bundleOutput\BankrbYaml.out 'CICS.CICS61.JCL(BANKRB)'
200 Port request OK.
125 Storing data set CICS.CICS61.JCL(BANKRB)
250 Transfer completed successfully.
ftp: 41738 bytes sent in 0.73Seconds 57.25Kbytes/sec.
ftp>
```

The commands file (BankrbYaml.out) that is built by the *zrb build* command is used as input to the CSD update batch utility program, DFHCSDUP, which updates the CSD data set with the new resource definitions. The updated CSD resource definitions are added to the CICS region.

Use the following JCL to add the resource definitions to CICS regions:

CICS.CICS61.JCL(DFHCSDIN)

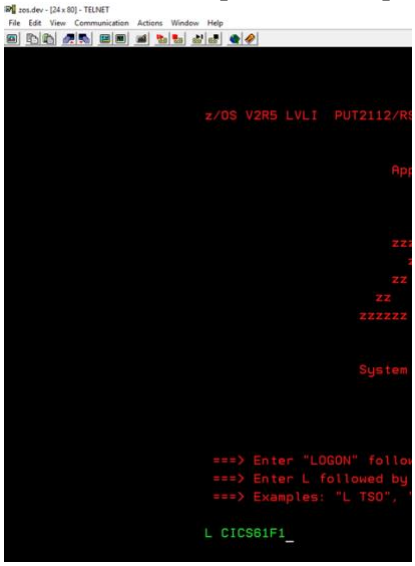
Part 6: Restart the CICS region (CICS61F1) and then, test the CBSA application in the CICS61F1 region

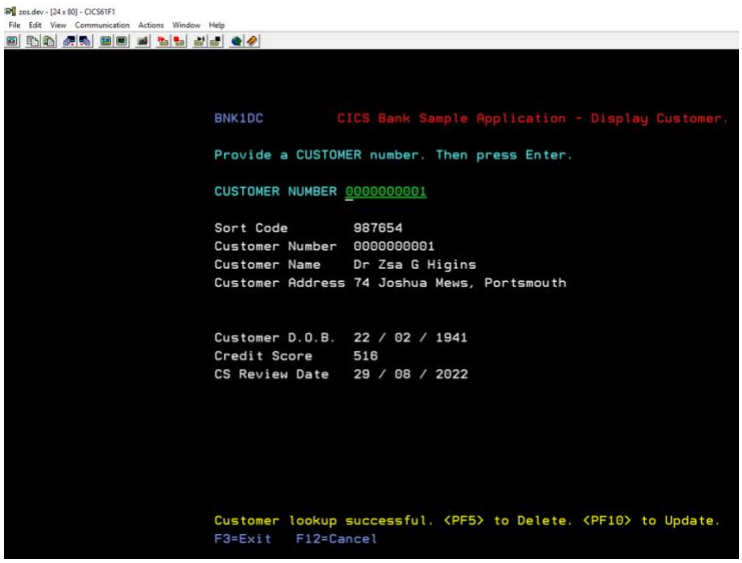
Restart the CICS regions in TSO/SDSF.

=S;DA
/s sto61
/s sta61

Issue “OMEN” transaction to test the CBSA application in CICS61F1 region.

L CICS61F1 ➔ [ESC / Clear] ➔ OMEN ➔ 1 ➔ 1





Now you can run the transaction in the CICS61F1 and have completed the lab successfully!!

Part 7: Summary

Congratulations, you have installed CICS CSD resources using YAML files

In this lab you performed the following steps:

- Extract CSD definitions from CICS region DFHCSD file.
- Import CSD definitions into a YAML file.
- Generate the DFHCSD commands to define the resources back into CICS region.
- Run the batch job to re-define the CICS resources with a different group name (BANKRB).