CBSA Installation Instructions – April 2022

Key: BLUE = Location on the mainframe

YELLOW = Location in the GitHub repo

GREEN = Existing location in USS

Set up libraries on the host machine and downloading data into them from GitHub:

The GitHub repo is on Enterprise GitHub and is called:

https://github.ibm.com/IBMZSoftware/CICS-Bank-Sample-Application

NOTE: CBSA V3 should be fully installed from scratch, and it replaces V2 or any previously installed CBSA instances (note there is no upgrade path between CBSA V2 and V3).

It is our intention to provide Ansible Playbooks to aid installation (for anyone wanting to install in this way). This will get added in due course.

Assumptions:

For Project Nazare, the supplied jobs should install straight into a zTrial environment. If you are not part of project Nazare and wish to install CBSA into your own environment, you may have to perform a small amount of tailoring in some of the jobs.

We have made some assumptions within this document, these are:

a. That a CICS region and the underlying libraries for that region are already installed in the host environment. Within this document, our CICS region on our test machine is called **CICSTS56.**

b. That a Db2 subsystem will be available in the host environment i.e. we will refer to the Db2 subsystem as **DBCG**.

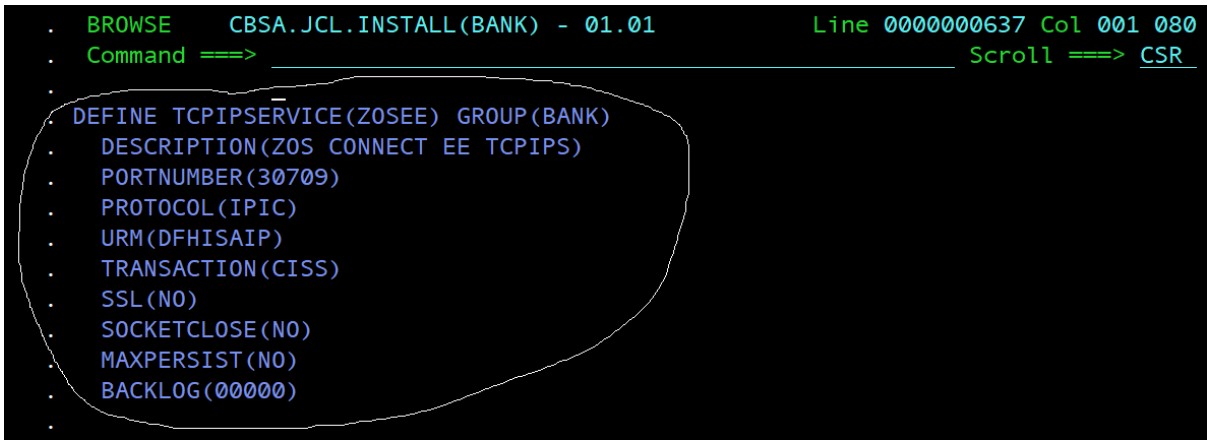
c. That there is already a zOS Connect EE Server installed. For the sake of illustration, ours is installed in USS at:
`/var/zosconnect/v3r0/servers/defaultServer/resources/zosconnect`

(if you are installing outside of project Nazare, this may be a different location in USS).

NOTE 1 – This installation will replace the 'default' zOS Connect EE **server.xml** with one which utilises the port numbers 30701 (HTTP) and 30702 (HTTPS) should these port numbers not be appropriate in your host environment then they may be changed in the **server.xml** member in the /zoseeserver folder.
NOTE 2 – The last line in **server.xml** references the connection between the z/OS Connect EE server and the CICSTS56 CICS system and it utilises port 30709:

```
<zosconnect_cicsIpicConnection id="cicsConn" host="localhost" port="30709" />
```

If this port number needs to be changed you will need to edit the server.xml in /zoseeserver and you will also need to edit the TCPIP Service definition ZOSEE, in member BANK in folder /jclInstall, to reflect the amended port number.



Installation instructions:

1. First, pull down, from Git Hub, the contents of the /jclInstall folder, on to your host machine (it needs to be put into a Library called CBSA.JCL.INSTALL).

This contains a set of jobs, which are used to create the necessary libraries and perform necessary set up.

2. Execute job CBSA.JCL.INSTALL (CRELIBS).

This job creates the following additional PDSE/LIBRARYs on the zTrial/host machine (which are all required later on):

PDSE/LIBRARY:	Description:
CBSA.DB2.JCL.INSTALL	– contains JCL jobs to set up DB2.
CBSA.CICSBSA.BUILDJCL	– contains the jobs (JCL) to compile all of the source code in CBSA.
CBSA.CICSBSA.COPYLIB	– contains the COPYBOOK members pulled into the SOURCE code.
CBSA.CICSBSA.LOADLIB	– contains the LOAD modules used by the CICSTS56 CICS region (these modules are created later during compilation).
CBSA.CICSBSA.DBRM	– contains the Db2 DBRM modules (which are generated by the bind).
CBSA.CICSBSA.LKED	– contains the Link Edit source entries.
CBSA.CICSBSA.BMS	– contains the BMS (map) source code modules

CBSA.CICSBSA.ASM	– contains Assembler source modules required to assemble DFHNCOPT
CBSA.CICSBSA.CBSAMOD	– contains the output from the COBOL compilation
CBSA.CICSBSA.COBOL	– contains the COBOL source code
CBSA.CICSBSA.DSECT	– The copylib that is used by installation compile jobs (i.e. not DBB compile. This folder contains Mapset DSECTs)
CBSA.CICSBSA.REORG	– contains the JCL for reorganising and changing the Db2 tables.

3. Once the CBSA.JCL.INSTALL(CRELIBS) job has completed and has created the above libraries, it will be necessary to pull down the content from the following GitHub folders into each respective library on zTrial.

Git Hub folder content	**->**	**zTrial LIBRARY**
/Db2_jcl_install		CBSA.DB2.JCL.INSTALL
/buildjcl		CBSA.CICSBSA.BUILDJCL
/copylib		CBSA.CICSBSA.COPYLIB
/linkedit		CBSA.CICSBSA.LKED
/bms		CBSA.CICSBSA.BMS
/asm		CBSA.CICSBSA.ASM
/cobol_before		CBSA.CICSBSA.COBOL
/reorg		CBSA.CICSBSA.REORG
/dsect_before		CBSA.CICSBSA.DSECT

Git Hub folder content	**->**	**USS location**
/aarfiles		Copy all members/files from /aarfiles into USS at /var/zosconnect/v3r0/servers/defaultServer/resources/zosconnect/**apis**/ (note - if your zOS Connect EE server runs from a different location in USS, copy the content of the /aarfiles folder into the location of your own …./resource/zosconnect/apis folder)

/sarfiles		Copy all members/files from /sarfiles into USS at /var/zosconnect/v3r0/servers/defaultServer/resources/zosconnect/**services**/ (note - if your zOS Connect EE server runs from a different location in USS, copy the content of the /sarfiles folder into the location of your own …./resource/zosconnect/services folder)

Git Hub folder content	**->**	**USS location**
/zoseeserver		This contains one member/file, the amended **server.xml** for the z/os Connect EE server. For project Nazare this needs to be copied over the top of the existing server.xml in USS (i.e. this completely replaces the existing server.xml) at location: /var/zosconnect/v3r0/servers/defaultServer (note - if installing into your own host environment (outside of project Nazare) you may wish to review the content of the **server.xml** and either replace it or merge it with your own server.xml)

Setting up the Db2 artefacts (STORGROUPS, TABLESPACES, TABLES, INDEXES):

The following creates the ACCOUNT, PROCTRAN and CONTROL table Db2 artefacts (The CONTROL table is utilised to store the last number in use).

1. Execute job CBSA.DB2.JCL.INSTALL(INSTDB2). This creates the Db2 artefacts e.g. STORGROUPS, TABLESPACES, TABLES and INDEXES (note at this time the ACCOUNT, PROCTRAN and CONTROL tables have no data on them yet).

Compile and BIND the source code modules:

1. Execute job CBSA.CICSBSA.BUILDJCL(COMPALL) to compile all of the source code.

2. Execute job CBSA.DB2.JCL.INSTALL(DB2BIND) to BIND the programs to Db2.

Create the VSAM files and populate the Db2 tables with data:

1. Execute job CBSA.JCL.INSTALL(BANKDATA) to create the ABNDFILE and CUSTOMER VSAM files and to execute program BANKDATA which populates the ACCOUNT and CONTROL Db2 tables with data (note the PROCTRAN table has no data in it to start with).

Set up the CICS region, SIT parms and the DFHCSD used to contain all of the CBSA definitions:

1. Execute job CBSA.JCL.INSTALL(REPLCICS). This copies member CBSA.JCL.INSTALL(CICSTS56) to FEU.Z24C.PROCLIB(CICSTS56) it is meant to replace/overwrite the provided CICS start up procedure (of the same name) in FEU.Z24C.PROCLIB. This sets up the correct DB2 libraries in STEPLIB and puts the CBSA load module into the DFHRPL.
If you are installing into your own host environment (outside of project Nazare) you may wish to review the content of member CICSTS56 and apply the relevant parts to your own CICS region startup procedure or startup JCL.

2. Execute job CBSA.JCL.INSTALL(REPLSIP). This copies member CBSA.JCL.INSTALL(DFH£SIP1) to DFH560.SYSIN(DFH£SIP1) it is meant to replace/overwrite the provided module (of the same name) in DFH560.SYSIN. This will setup the correct SIT parameters (as used by the CICS region) and includes a Db2 connection.

If you are installing into your own host environment (outside of project Nazare) you may wish to review the content of member DFH£SIP1 and apply the relevant parts to your own SIT parameters.

3. Execute job CBSA.JCL.INSTALL(HBANKCSD) to update the DFHCSD with the CBSA definitions and add the CSD GROUP(BANK) into a LIST called CICSTS56 (which is used in the SIT parms for the CICS region).

Make RACF and USS security changes

If you are installing into your own host environment (outside of project Nazare) you may wish to review the content of members RACF001 and ZOSCSEC and apply the relevant parts to your own environment.

1. Execute job CBSA.JCL.INSTALL(RACF001). This makes the RACF changes to ensure that the new IBMUSER controlled CICS region has access to everything when it restarts (in particular the Db2 sub system, since we have added DB2CONN into the SIT parm previously).

2. Execute job CBSA.JCL.INSTALL(ZOSCSEC). This makes the security change to ensure that the APIs and Services for the RestFul APIs, used by zOS Connect EE, have the correct access when the z/OS Connect EE server is restarted later on.

Stop and Restart the CICS region.

In order to pick up the changes to the SIT parms and the new CSD definitions for CBSA, it is necessary to stop and restart the CICSTS56 CICS region (the CICS region being utilised is defined with **START=INITIAL** coded in the SIT, so restarting CICS pulls in all of the above changes).

1. Execute job CBSA.JCL.INSTALL(SHUTCICS) to close down the CICS system (CICSTS56).

2. Execute job CBSA.JCL.INSTALL(RESTCICS) to start up CICSTS56 (this time pulling in the SIT parm amendments etc.).

Stop and Restart the zOS Connect EE Server.

In order to pick up the changes to the z/OS Connect EE Server (which may already be running):

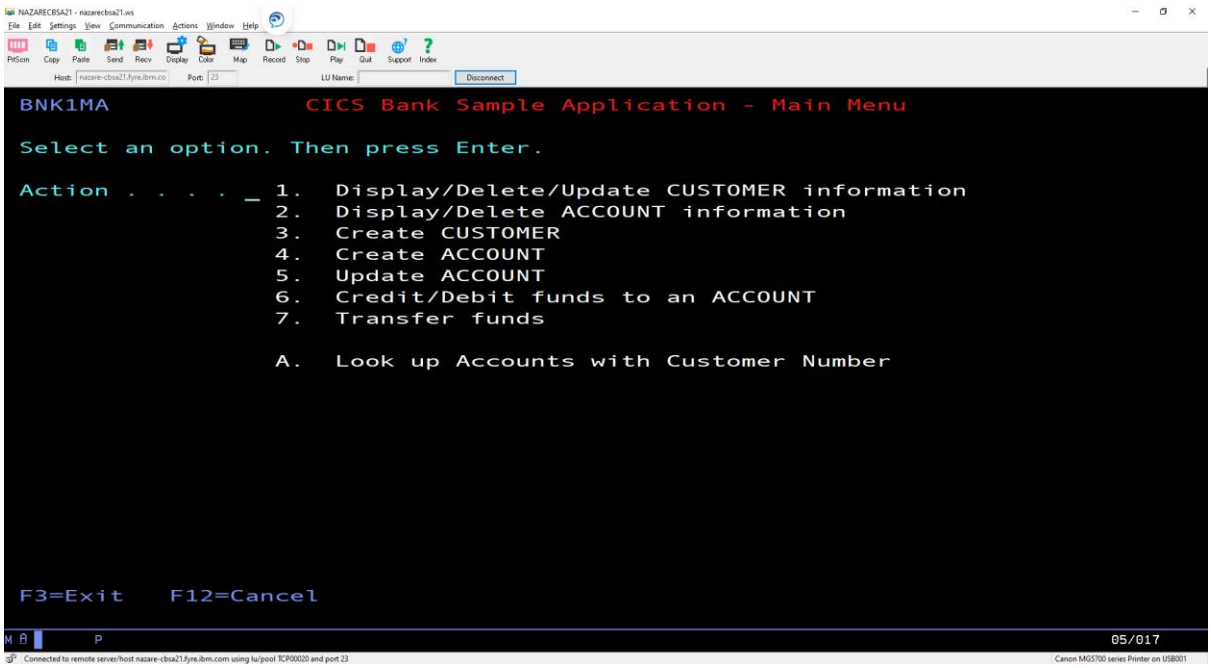
1. Execute job CBSA.JCL.INSTALL(SHUTZOSC) to close down the z/OS Connect EE server.

2. Execute job CBSA.JCL.INSTALL(RESTZOSC) to restart the z/OS Connect EE server.

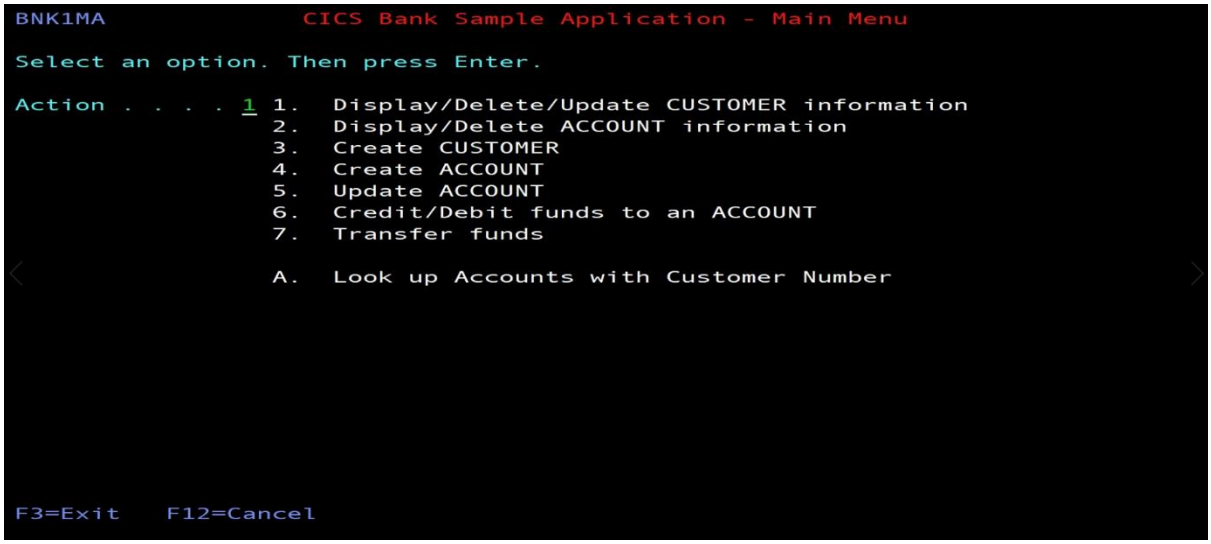
Validating that the installation has worked:

Checking the CUSTOMER VSAM KSDS:

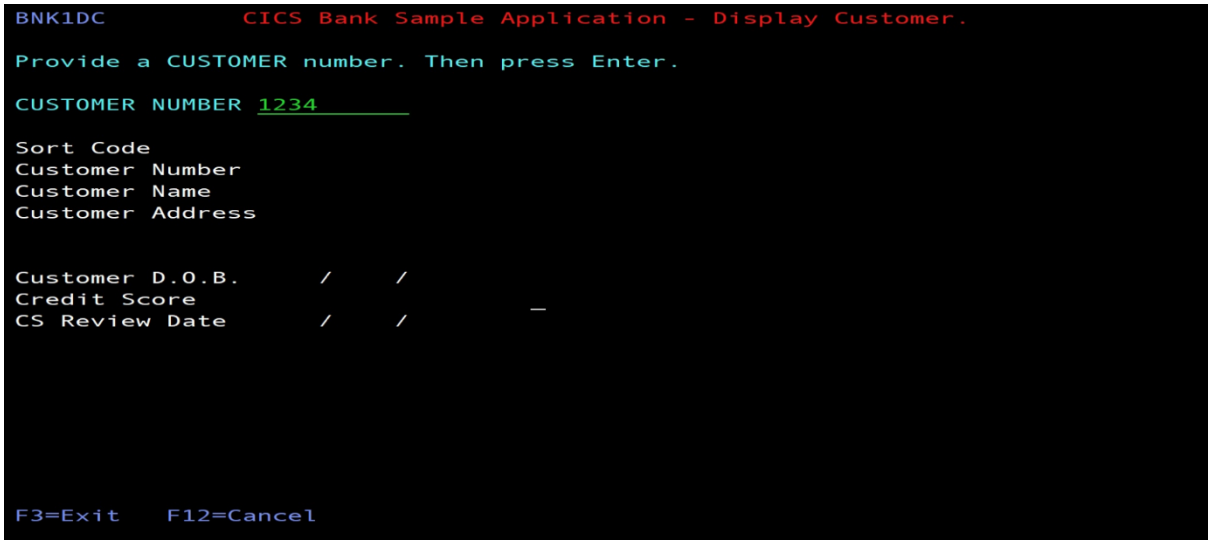
1. Logon to CICSTS56 (with the IBMUSER supplied USER ID), Clear the screen, Type OMEN and you should be presented with the CBSA MAIN MENU:



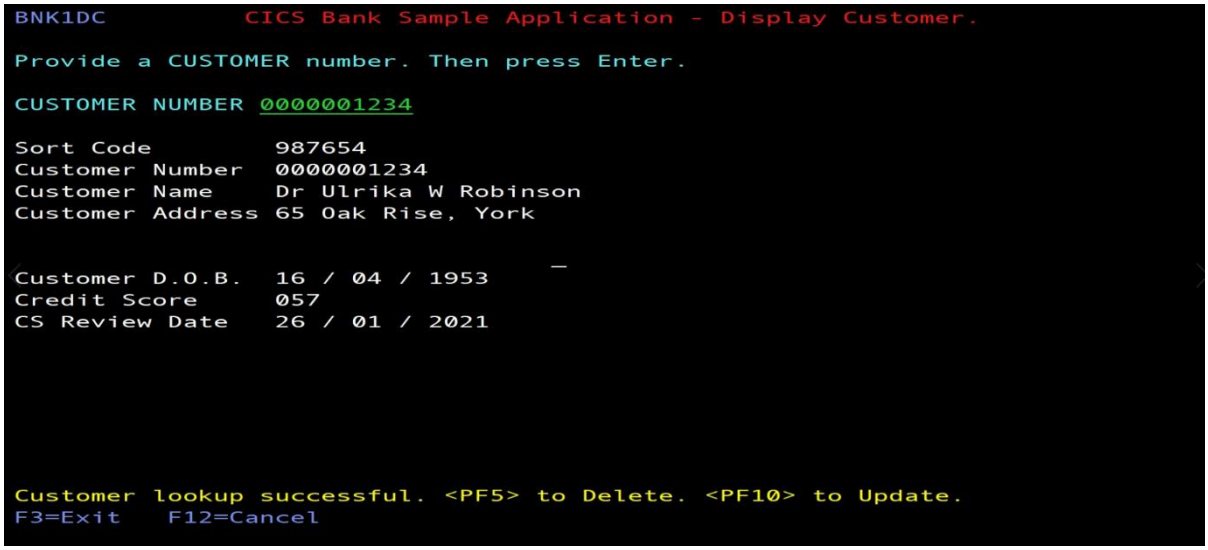
2. Select option 1 (Display/Delete/Update CUSTOMER information) from the main menu:



Then enter a Customer Number:



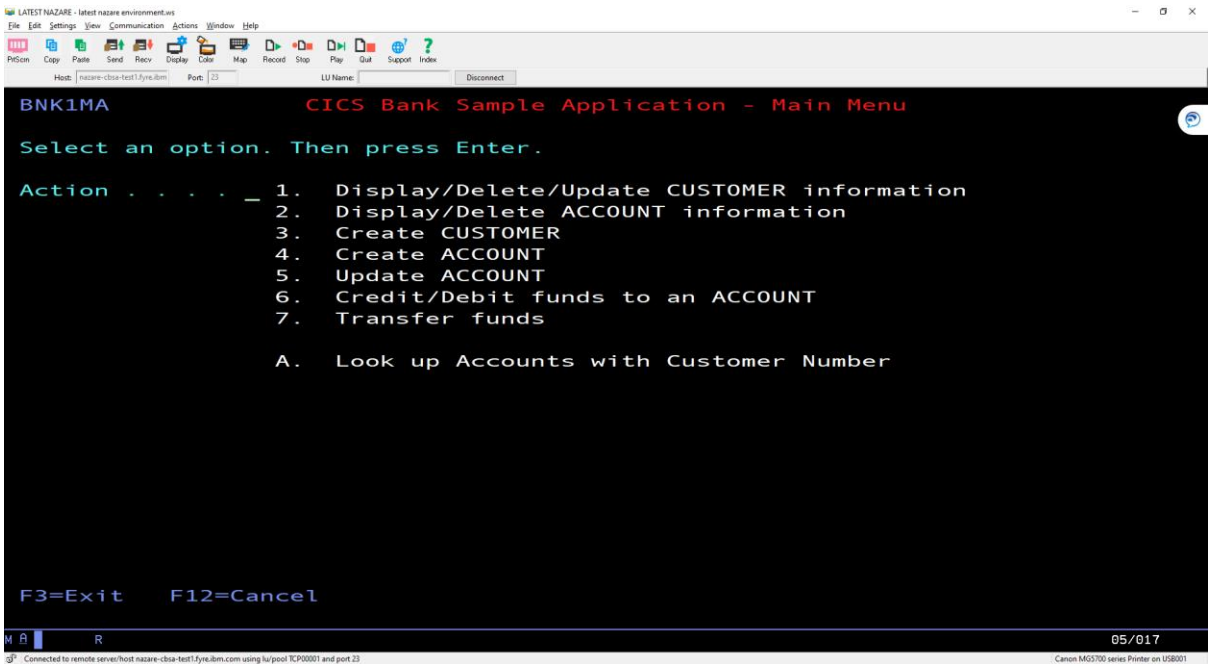
If all is well, the details for that particular customer should be returned:



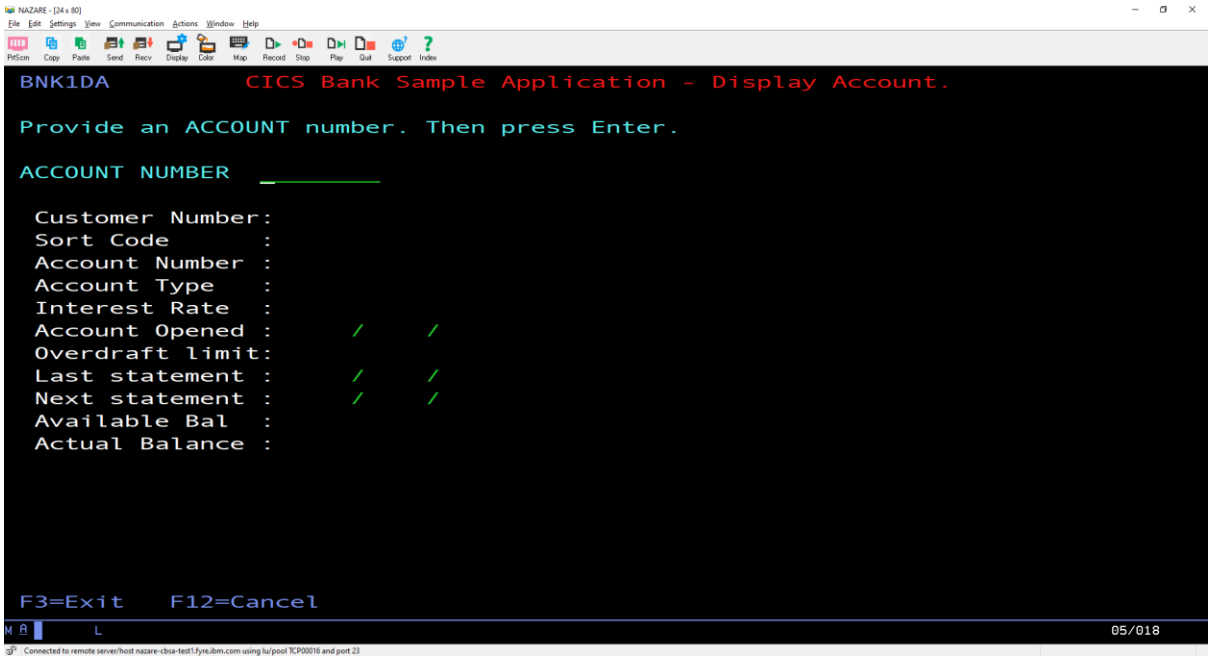
(this validates that KSDS data has been loaded correctly).

Checking the ACCOUNT data (Db2):

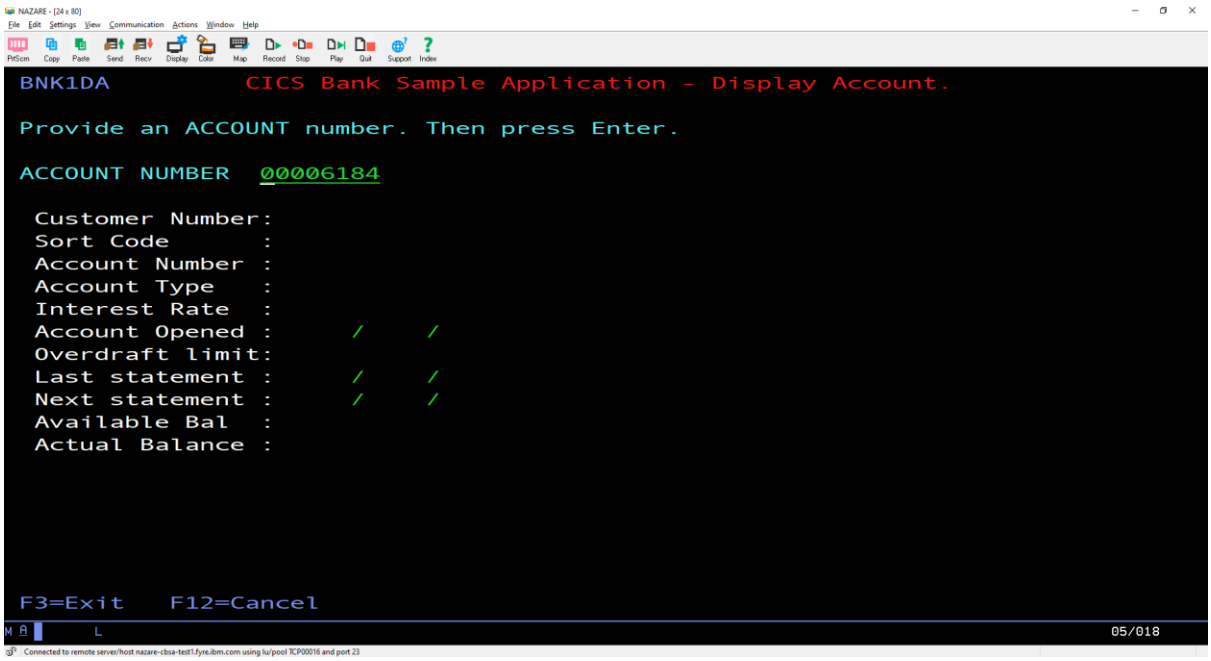
3. Go back to the CBSA main menu (pf3):



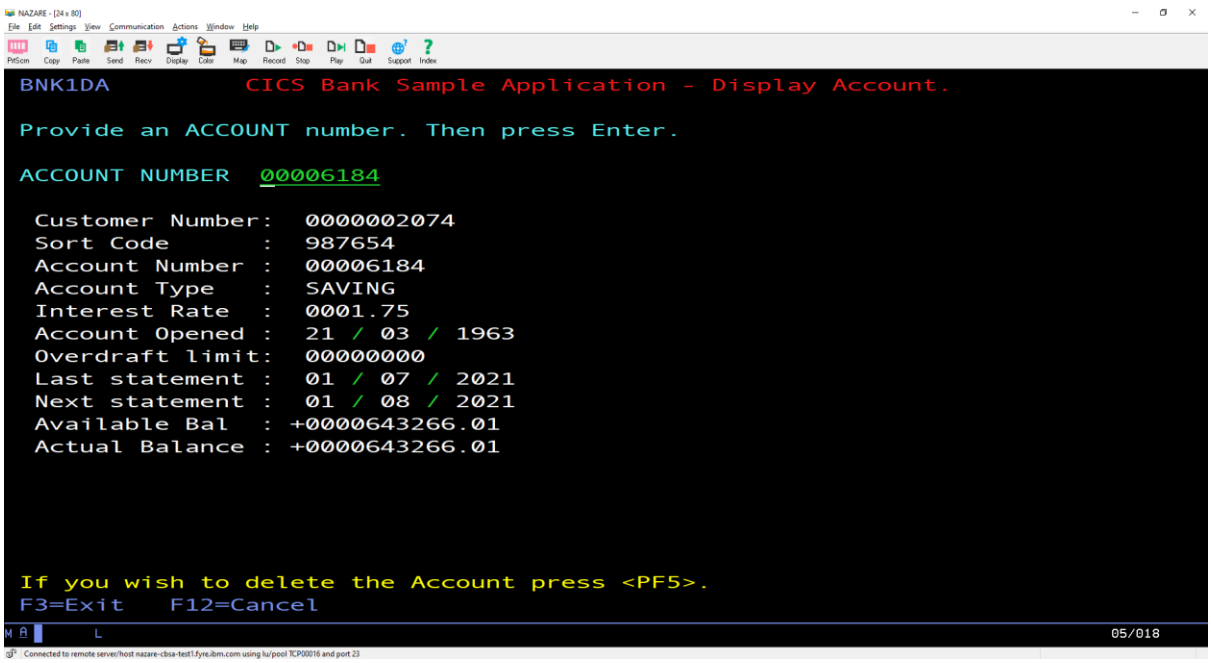
And from here select option 2 (Display/Delete ACCOUNT information):



The account number field should be 8 bytes long. Supply an account number, for example:



And this should return some data:



If data is displayed, this validates that the Db2 ACCOUNT table has been correctly populated.

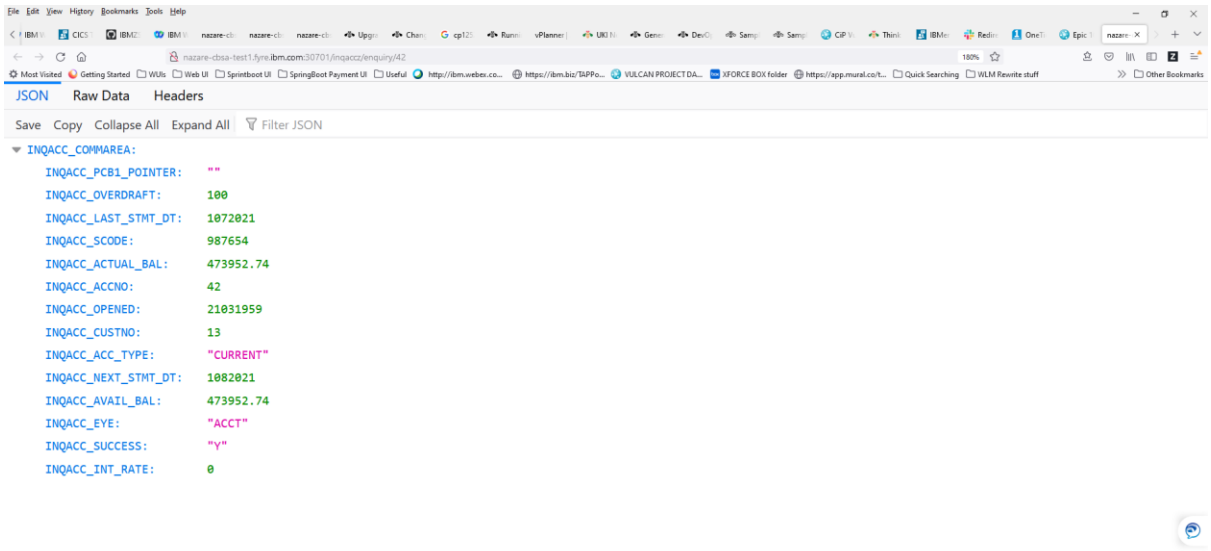
Checking the zOS Connect EE API:

4. Confirm that the zOS Connect EE server is executing (check this in SDSF (S;DA)) – the job name for the zOS Connect EE server in project Nazare is called **ZOSCSRV**.

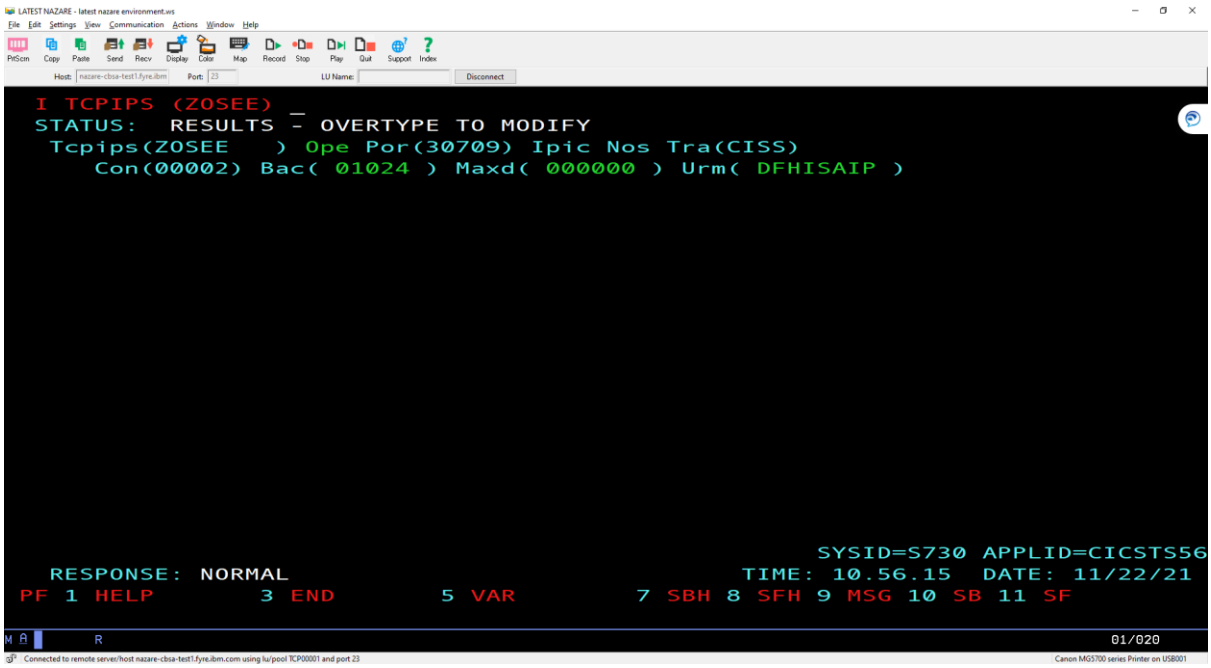
5. Go to a web browser and put in this URL:

> http://*your-host-name*:*your port-number*/inqaccz/enquiry/42
>
> For project Nazare, replace *your-port-number* with port 30701 in the above URL. You will have been allocated your own zTrial hostname if running as part of project Nazare, and this should replace *your-host-name* in the above URL.
>
> If the zOS Connect EE server is executing outside of the zTrial/Nazare environment, substitute the host name and port number accordingly.

6. This should return something similar to this (enquiry on account 42):



If data is not returned, then check the port number and status of the z/OS Connect EE server. If these are OK, check in the CICS region (using the **CEMT I TCPIPS(ZOSEE)** command) that there is a TCPIP Service called **ZOSEE** and that it has been installed and that the TCPIP Service is Open. You should see something like this:



The above TCPIP Service should be using port 30709 - this is the connection between the zOS Connect EE server and the CICST56 CICS region supplied in zTrial.

Finally, check the zOS Connect EE Server logs to ensure that the API archive files (the aars) and the service archive files (the sars) were all installed successfully and that the server itself is reporting a message similar to this:

> The defaultServer server is ready to run a smarter planet. The defaultServer server started in 112.474 seconds.