

# IBM TechU

2021 Edition



IBM®

Lab Exercise:

**Using IBM Z VTP to test a COBOL CICS DB2 transaction (Portuguese/Spanish).**

Session ID:

**z203785**

Speaker Name:

**Reginaldo Barosa**

Room:

**?????...**

**FOR LAB ROOM ONLY -- DO NOT REMOVE, DEFACE or WRITE ON THIS LAB BOOK**

## Introduction

This lab will take you through the steps of using the automated testing capabilities of the **IBM Z Virtual Test Platform (VTP)** to create a test at the transaction levels of a COBOL/CICS/ DB2 application. This is done by stubbing out CICS, IMS, DB2, DL/1 and MQ calls, enabling the program to be tested without a DB2 and CICS environment being active. In this lab you will record the transaction using an existing COBOL/CICS/DB2 application and write the recorded data into a flat file (playback file). Then you will make a change to the COBOL/CICS/DB2 program adding a bug and rerun the test that should catch the bug.

### Instructions to use the Windows + Linux + Z/OS on Cloud

There are 2 ways to access the Windows + z/OS + Linux cloud instances..

1. Via **Windows Remote Desktop** (Better performance, but firewalls might block.)
2. Via **Web Browser** (suggested Firefox or Chrome)

In any case be sure that you have the **provided link to access the cloud instance**, the userid and **password** to access the Windows client remotely.

Please always **Use COPY/PASTE for password**.

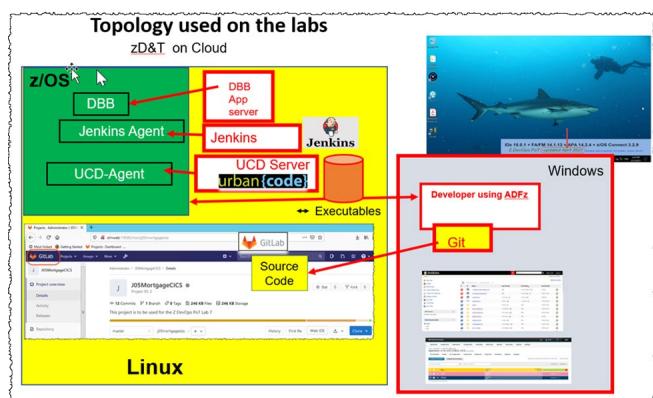
Some letters may be identical, example: **I** (uppercase i) and **l** (lowercase L).

Example of provided links

wIBM Z System DevOps Workshop		User ID for Web browser	IP Address for Remote Password	User ID for Remote Desktop	Domain	machine #
Serial No	URL	Administrator	169.63.141.246	K5BE6Pep	Wwin-5282\Administrator	Wwin-528
1	<a href="https://169-63-141-246-T-5282.ibmtrialmachines.com/">https://169-63-141-246-T-5282.ibmtrialmachines.com/</a>	Administrator	169.60.90.69	FBtbmGI5	Wwin-5283\Administrator	Wwin-5283

Below is what you will access using the cloud environment.

Be sure that you have an IP address, the userid ( **Userid will be different if using Remote Desktop or Browser** ) and password to access the Windows client remotely.



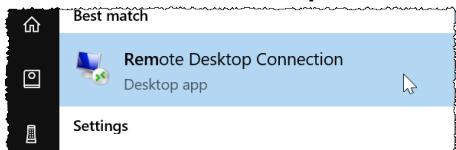
## Using Windows Remote Desktop (preferable way)

This is the preferable way to run the labs, but in some customer location this capability is blocked via Firewalls. If this is your case, use the Web Browser. Instructions are listed here as well..

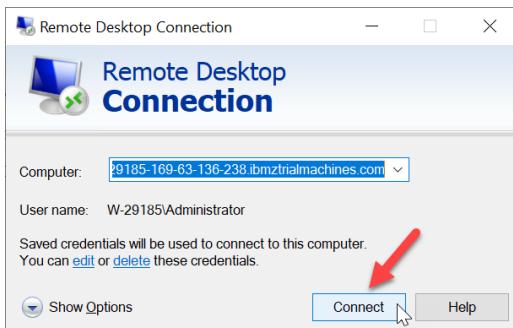
If you are a Mac user may consider downloading the Remote desktop from  
<https://itunes.apple.com/us/app/microsoft-remote-desktop-10/id1295203466?mt=12>

1. Here one example using Windows 10.

Start the **Remote Desktop** .

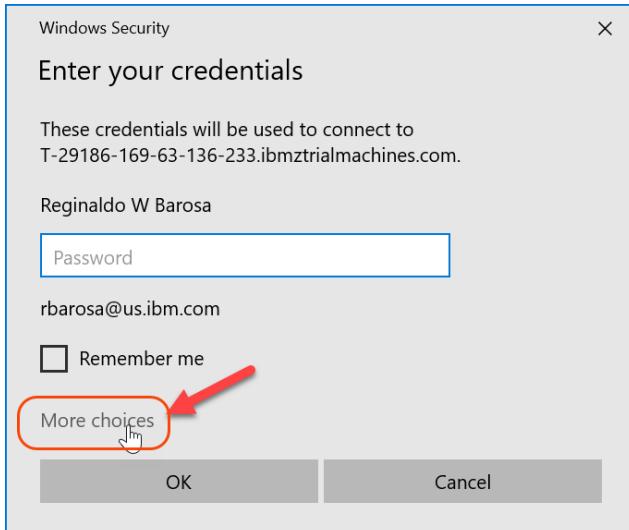


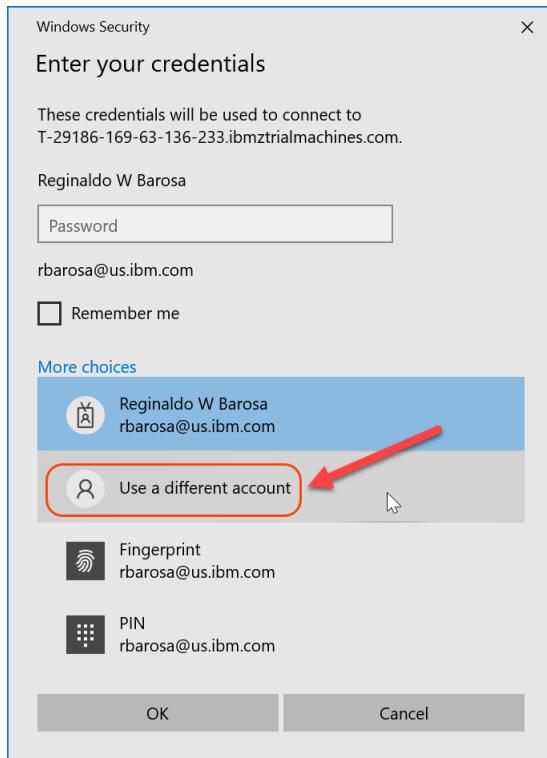
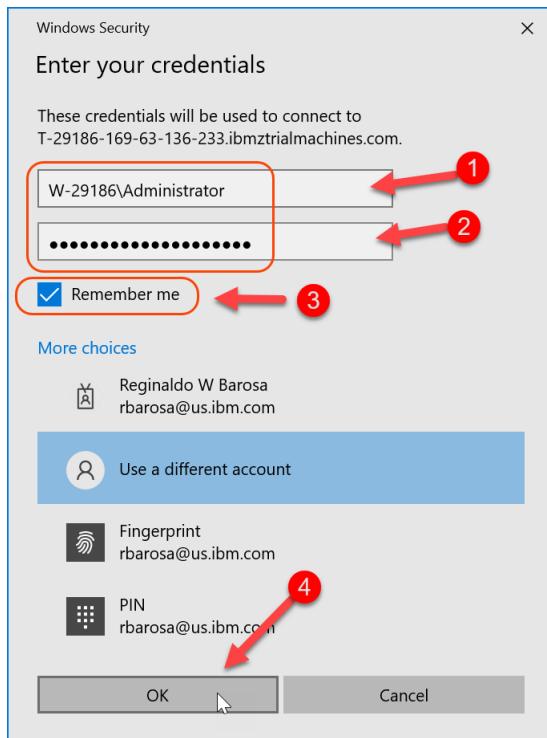
2. Click **Connect**



3. Click **Connect** if a dialog as to *trust the remote connection*

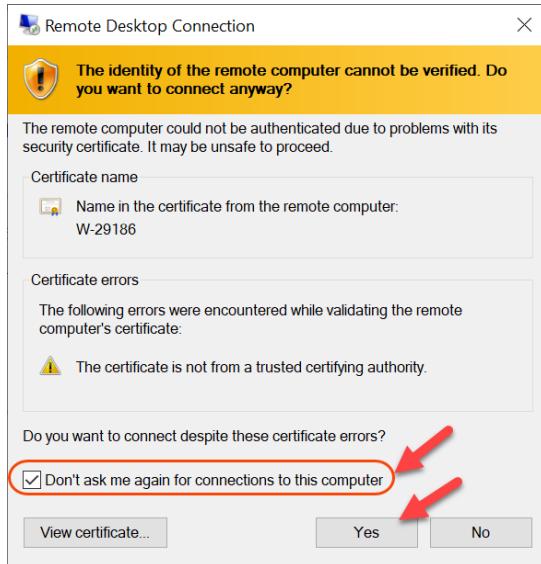
4. Click **More Choices**



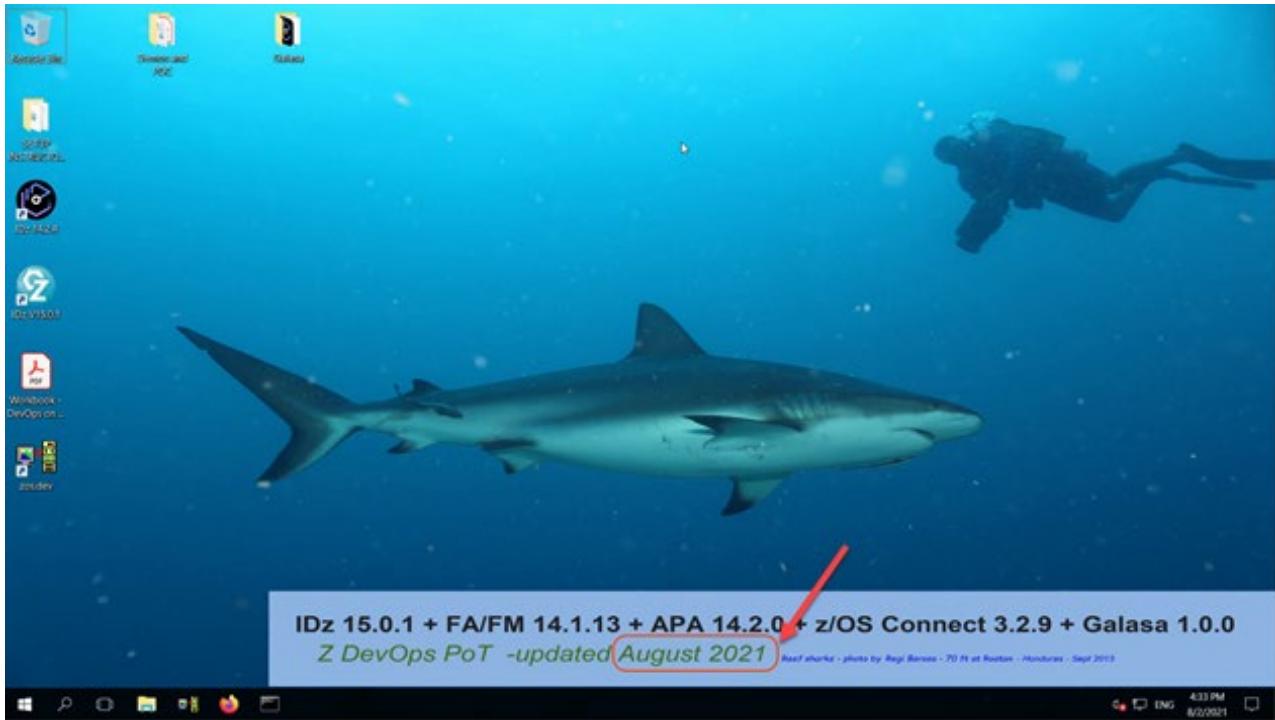
**5. Click Use a different account****6. Copy and paste the provided “User ID for Remote Desktop” and **Password**, select **Remember me** and click **OK**.**

7. You should get a dialog as below..

Select Don't ask me again for connections to this computer and click Yes



8. You should get a screen with a shark. That indicate that you have access to the Windows client on cloud.



On the windows desktop there is a PDF icon with the Labs workbook.. you will find this lab also there.  
It may help for copy/paste, etc...

But to follow the labs I suggest printing or use another display or iPhone/iPad to better follow the instructions or use another Monitor to follow the lab instructions.

## Using Web Browser (if using VPN or Firewalls that block Remote Desktop)

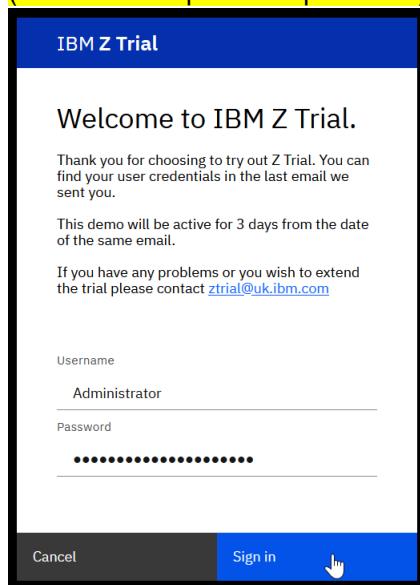
Be sure that you have the **provided link to access the cloud instance**, the userid (**Administrator**) and **password** to access the Windows client remotely. Always use **COPY/PASTE** for password.

Some letters may be identical, example: **I** (uppercase i) and **l** (lowercase L).

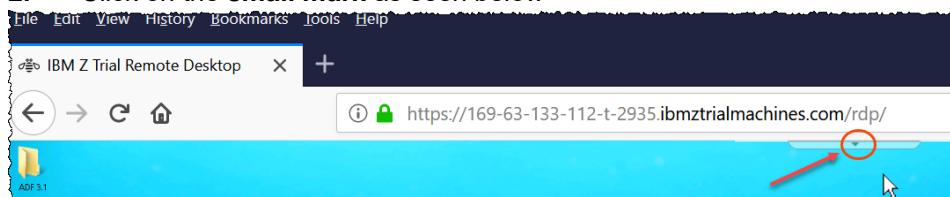
### Example

ibm Z System DevOps Workshop		User ID for Web browser	IP Address for Remote Password	User ID for Remote Desktop	Domain	machine #
Serial No	URL	Administrator	169.63.141.246	KS8E6Pep	Wwin-5282\Administrator	Wwin-528 5282
	1 https://169-63-141-246-T-5282.ibmtrialmachines.com/	Administrator	169.60.90.69	F8thmGL5	Wwin-5283\Administrator	Wwin-528 5283
	2 https://169-60-90-69-T-5283.ibmtrialmachines.com/	Administrator				

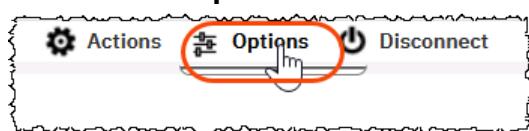
1. Use the link provided by the instructor and type **Administrator/password** provided and click **Sign in** (**Remember to paste the password**)



2. Click on the **small mark** as seen below



3. Click on **Options**

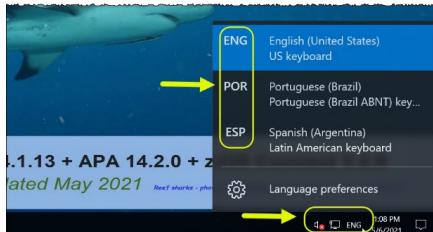


And select **Full Screen**

## Adjusting the keyboard for other languages than English

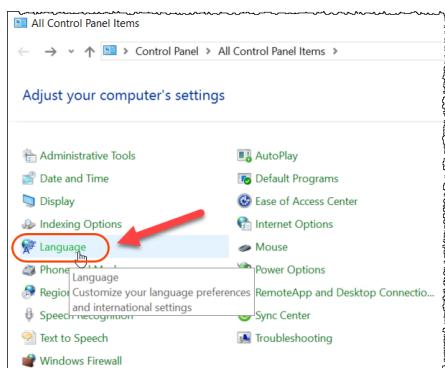
In some countries the keyboard must be mapped due language differences.

If you are in Brasil or any other Latin America country can select the language as below

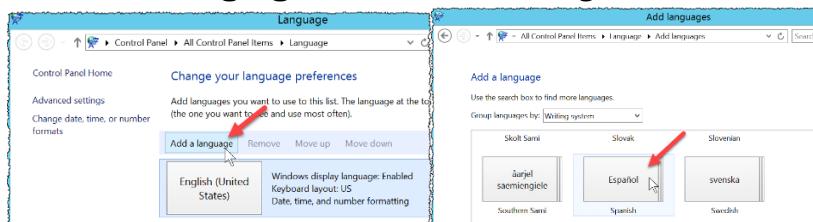


If you use another keyboard other than ENG, POR or ESP, you need to make updates..

1. Go to windows **Control Panel** and select **Language**



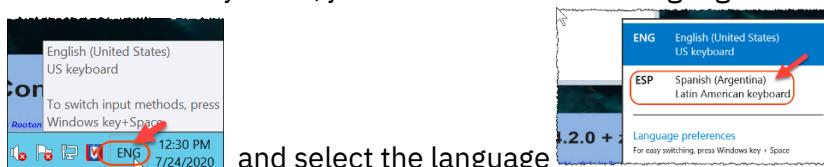
2. Click **Add a Language**, and follow the dialogs. See one example on the screen captures below



**Click Add**



To switch the keyboard, just select the desired language on the right corner of the screen



and select the language

## Exercise instructions

### Overview of development tasks

To complete this tutorial, you will perform the following tasks:

#### PART #1 – Record existing COBOL/CICS/DB2 transactions and introduce a bug

1. **Use VTP to record the CICS/DB2 application in action**  
→ You will record the CICS application in action. This application starts multiple CICS transactions using COBOL programs with and without DB2 access.
2. **Run a JCL to execute the recorded replay sequence**  
→ You will submit a JCL to execute the sequence of interactions that you recorded and verify the results.
3. **Modify one COBOL/DB2 program (introduce a bug) and rerun the VTP JCL**  
→ You will modify the COBOL/DB2 program to introduce a bug, rerun the replay JCL, and observe the failure due the introduced bug.

#### PART #2 – Fix the program bug and re-run the test

4. **Run the CICS transaction and verify the bug**  
→ You will run CICS application and observe the bug introduced.
5. **Use IDz to fix the bug, recompile the COBOL/DB2 program**  
→ The bug is fixed using IDz, program fixed is rebuilt.
6. **Rerun the VTP JCL and verify that the bug is eliminated**  
→ When running the JCL recorded replay you can verify that the program is fixed.

## PART #1 – Record existing COBOL/CICS/DB2 transactions and introduce a bug

### Section 1. Use VTP to record the CICS/DB2 application in action

You will start a 3270 emulation and execute a transaction named **HCAZ** to record some of the interactions using VTP.

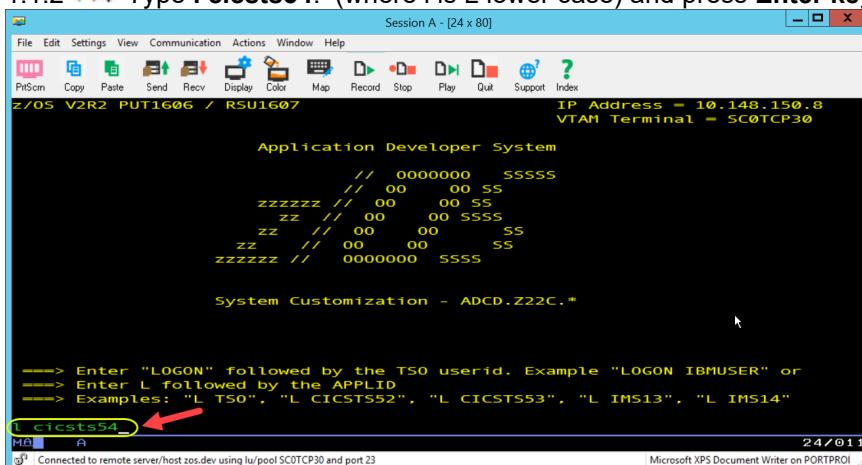
#### 1.1 Emulate a 3270 terminal and Connect to CICS version 5.4

1.1.1 Bring up a 3270 terminal emulator clicking on the **host emulator icon** on the Windows task bar.

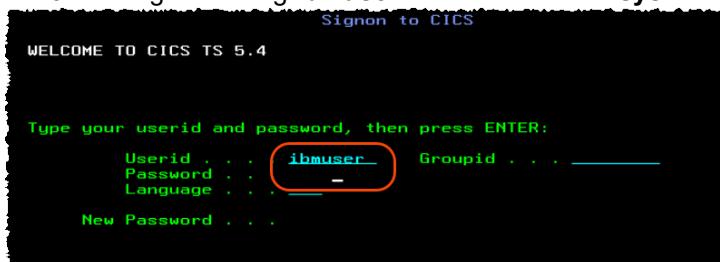


This opens the host emulator.

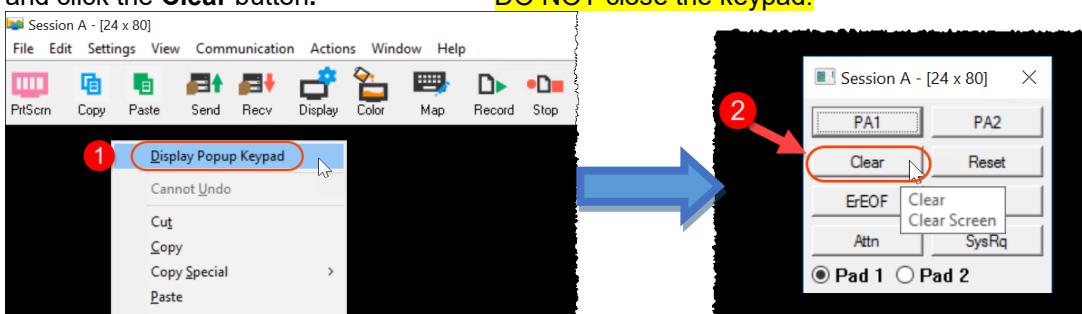
1.1.2 Type **I cicsts54.** (where I is L lower case) and press **Enter key.**



1.1.3 Sign on using **ibmuser** as the userid and **sys1** as the password.

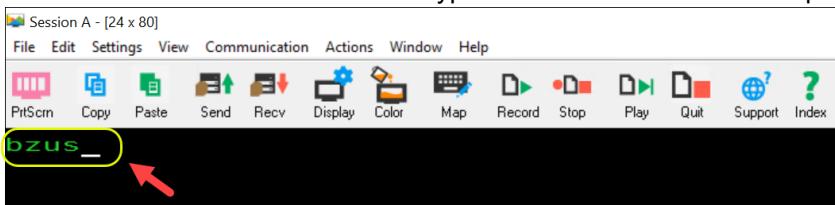


1.1.4 To clear the screen, **right click** on the dark space and select **Display Popup Keypad** and click the **Clear** button. **DO NOT close the keypad.**

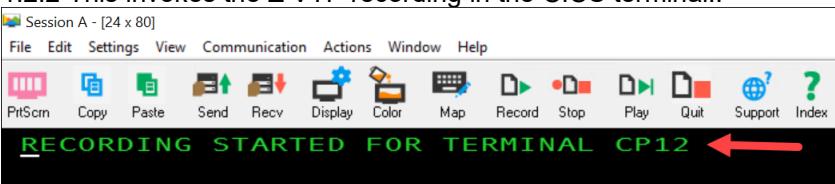


## 1.2 Using VTP to record the Health Care Application transaction sequence

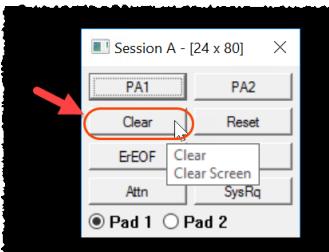
- 1.2.1 Click on the black area to type the transaction **bzus** and press Enter key



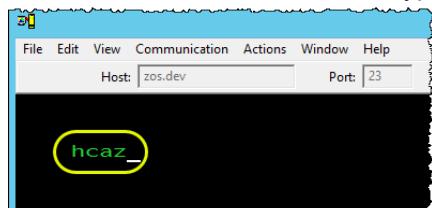
- 1.2.2 This invokes the Z VTP recording in the CICS terminal..



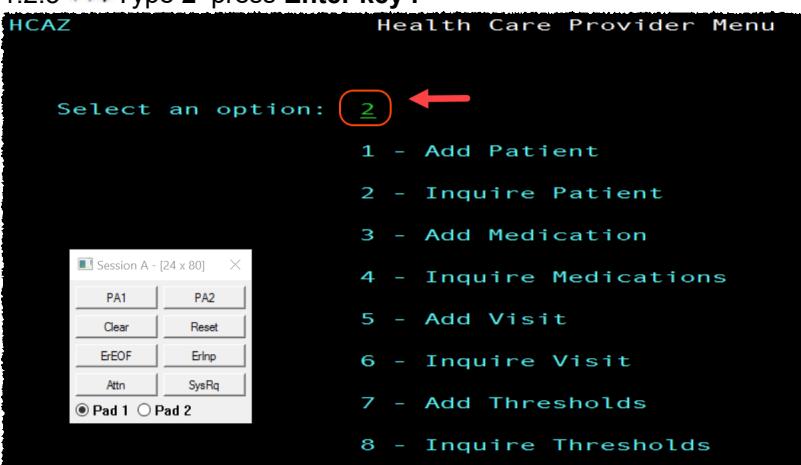
- 1.2.3 Clear the screen again clicking the **Clear** button.



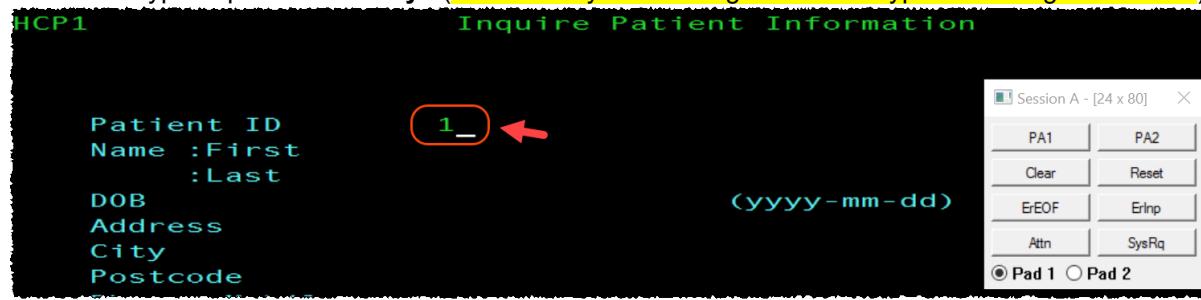
- 1.2.4 On CICS main terminal type **hcaz** and press **Enter key** to invoke the CICS transaction application



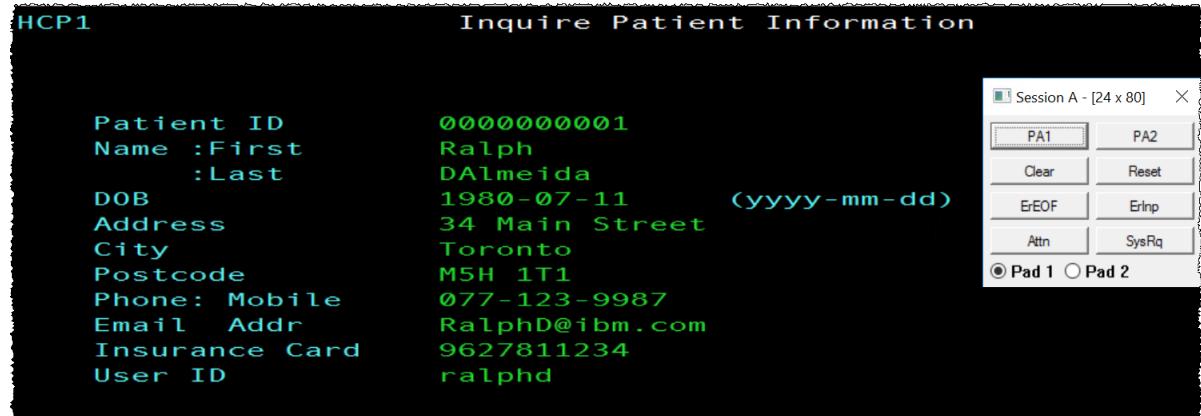
- 1.2.5 Type 2 press **Enter key** .



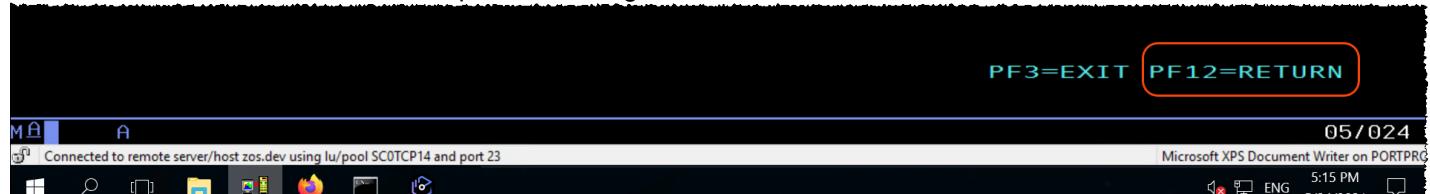
1.2.6 ► Type 1 press **Enter key** . (Notice that you can drag the stick keypad to the right if needed)



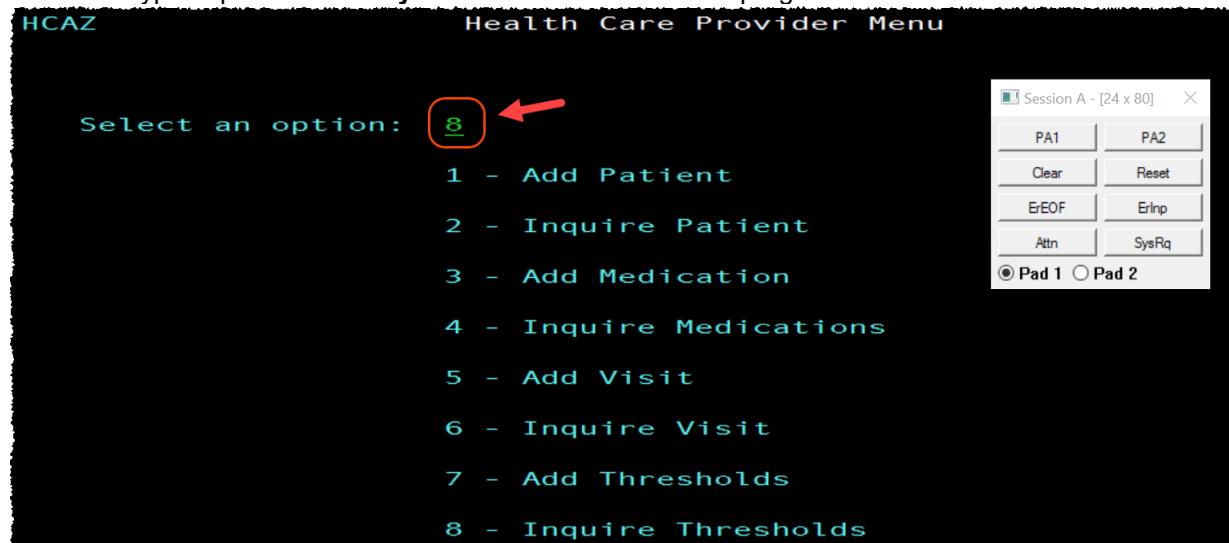
You should have the patient data displayed.



1.2.7  Press **PF12** to return to the previous dialog



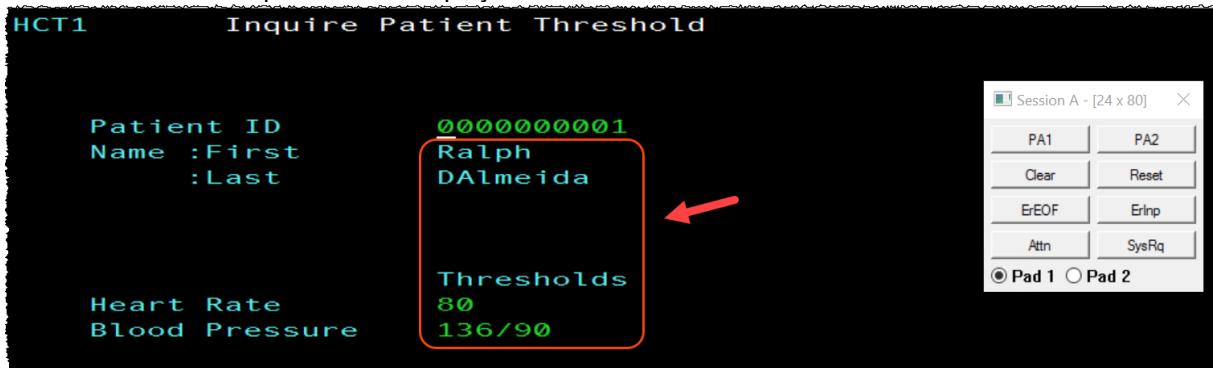
1.2.8 ➡ Type **8** press **Enter key** . This will invoke different programs



1.2.9 Type 1 press Enter key . (Notice that you can move the stick keypad to the right if needed)

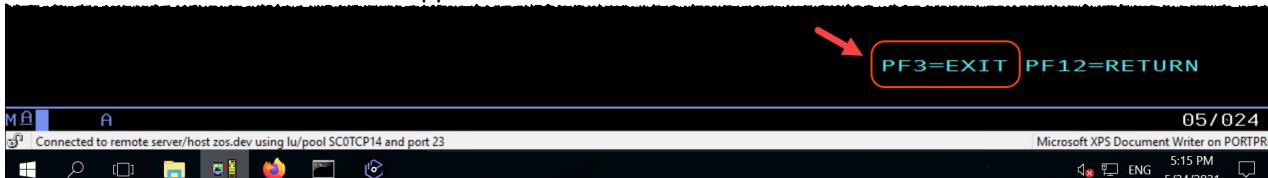


You should have the patient data displayed.



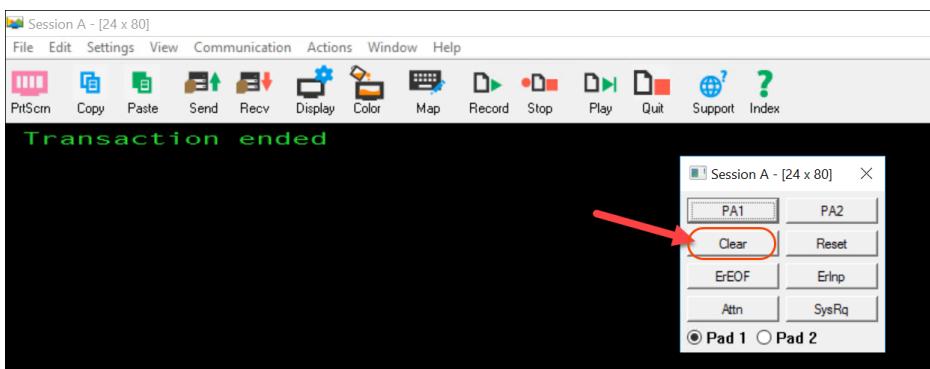
**IMPORTANT:** Remember the *first name* of this patient (Ralph) Later you will introduce a bug.. Instead of **Ralph** the COBOL program will display “**BAD NAME**” as first name.

1.2.10 Press PF3 to end the application



1.2.11 After the Transaction ended, use the key pad and click **Clear**

It will clear the screen and return to the main CICS terminal..



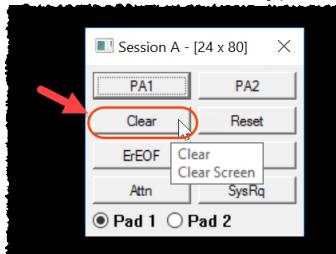
1.2.12 Type the transaction **bzue** and press **Enter key**



1.2.13 ► The following message appears in the screen.



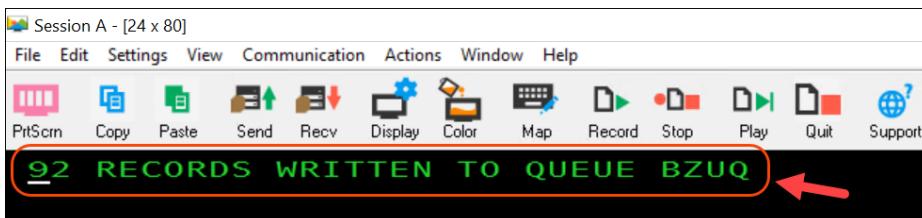
1.2.14 ► Use the Keypad to click on the **Clear** button and return to the main CICS terminal.



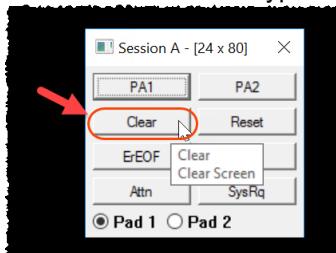
1.2.15 ► Type the transaction **bzuw** and press **Enter key**



1.2.16 This captures the recording into a CICS Transient Data Queue (TDQ) called **BZUQ** as displayed in the message.



1.2.17 ► Use the Keypad to click on the **Clear** button and return to the main CICS terminal.



1.2.18 You need to close this CICS Queue to be used in a batch job in next steps.

► Type **CEMT INQ TD** and press **Enter key**.



1.2.19 This TDQ is opened.

► Using the Tab key move the cursor under the letter "O" on the **BZUQ** queue.

```
INQ TD
STATUS: RESULTS - OVERTYPE TO MODIFY
Tdq(BZUC) Ext           Ena Clo
                         Shr Inp   Dat(001) Ddn(BZUCFG )
Tdq(BZUQ) Ext           Ena Ope   Old Out   Dat(001) Ddn(BZUBZUQ )
                         Shr Inp   Dat(001) Ddn(BZUCFG )
Tdq(CADL) Ind Nam(CSSL)
```

1.2.20 ► Type **C** and press **Enter key**. Notice the message **NORMAL**

This file is now closed and can be used on a batch job.

```
INQ TD
STATUS: RESULTS - OVERTYPE TO MODIFY
Tdq(BZUC) Ext           Ena Clo
                         Shr Inp   Dat(001) Ddn(BZUCFG )
Tdq(BZUQ) Ext           Ena Clo   Old Out   Dat(001) Ddn(BZUBZUQ )
                         Shr Inp   Dat(001) Ddn(BZUCFG )
Tdq(CADL) Ind Nam(CSSL)
```

1.2.22 You need to know which is the z/OS data set name that you closed..

► Using the tab key move the cursor on the **Tdq(BZUQ)** line and press **Enter Key**.

```
INQ TD
STATUS: RESULTS - OVERTYPE TO MODIFY
Tdq(BZUC) Ext           Ena Clo
                         Shr Inp   Dat
Tdq(BZUQ) Ext           Ena Clo   Old Out   Dat
                         Shr Inp   Dat
```

1.2.23 You will see the Data set to be used in the VTP JCL that holds the recording.

This dataset is named **BZU100.ZUNIT.PLAYBACK**

```
INQ TD
RESULT = OVERTYPE TO MODIFY
Tdqueue(BZUQ)
Type(Extra)
Nameind()
Triggerlevel()
Enablestatus( Enabled )
Openstatus( Closed )
Termid()
Tranid()
Userid()
Disposition(Old)
Iotype(Output)
Indoubt()
Indoubtwait()
Databuffers(001)
Ddname(BZUBZUQ)
Dsname(BZU100.ZUNIT.PLAYBACK) ←
Member()
+ Installtime(05/20/21 14:45:23)
```



You have now successfully created a VTP test case. It is a good practice to copy this data set into another file. In case you made mistakes on this lab on the recording we kept

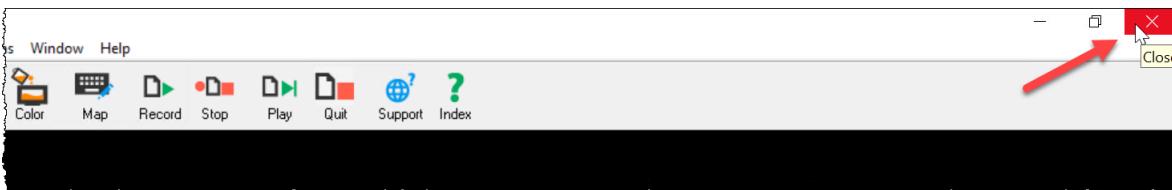
1.2.24 ► Press **PF3** to end this dialog



1.2.25 It will end the dialog



1.2.26 ► Close the terminal emulation



#### What have you done so far?

You executed the CICS transaction **HCAZ** and using VTP you recorded a simple interaction with the Health Care application. The recorded data is saved on a Z/OS dataset..

## Section 2 – Run a JCL to execute the recorded replay sequence

Using IDz you will submit a JCL to execute on z/OS that will verify the sequence recorded by VTP against the COBOL/CICS/DB2 programs that are being used on CICSTS5.4.

### 2.1 Connect to Z/OS using IDz

2.1.1 Start *IBM Developer for z Systems version 15* if it is not already started

► Using the desktop double click on **IDz V15** icon.

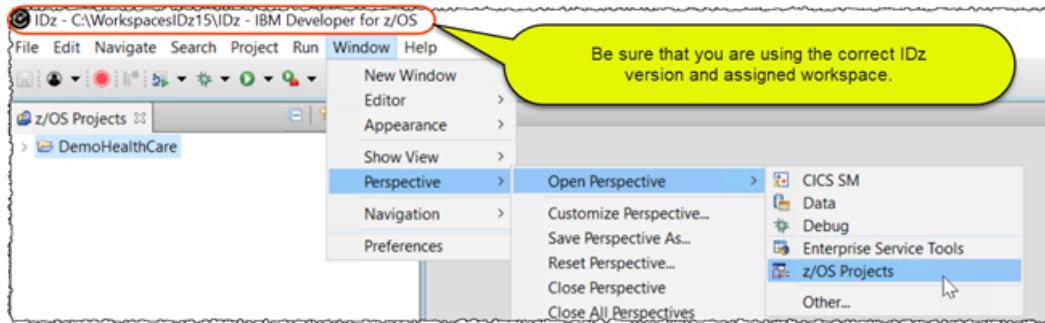
► Verify that the message indicates that it is Version **15.0.1**

**IMPORTANT** -> This icon will start an eclipse workspace that has already some definitions required for this lab.  
PLEASE DO NOT start IDz using other way than this icon.



## 2.2 Submit a provided VTP JCL for execution

- 2.2.1 Open the z/OS Projects perspective by selecting  
Window > Perspective > Open Perspective > z/OS Projects

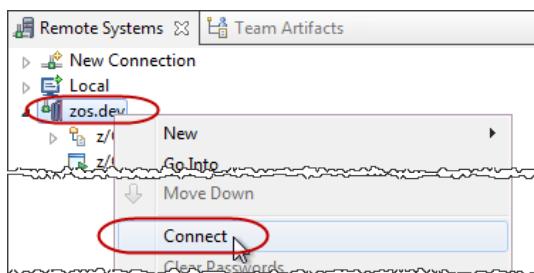


If nothing happens is because you are already at this perspective.

- 2.2.2 On this lab you will use userid **ibmuser**. and password **sys1**.

If you are connected as **ibmuser**, jump to step 2.2.4 Otherwise.

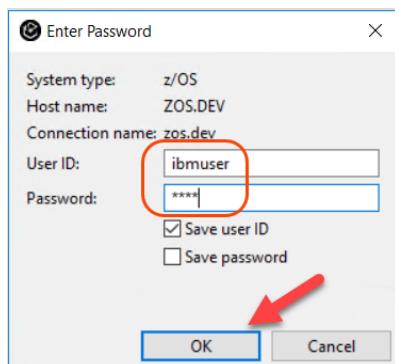
- Using Remote Systems view, right click on **zos.dev** and select **Disconnect** and then **Connect**



- 2.2.3 Type **ibmuser** as userid and **sys1** as password.

The userid and password can be any case; don't worry about having it in UPPER case.

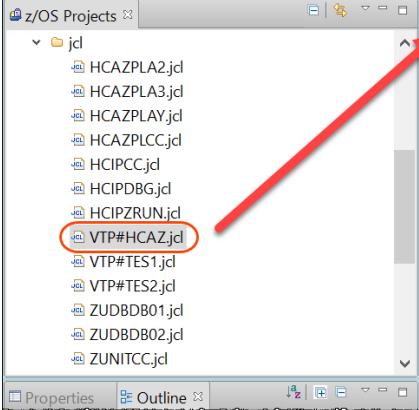
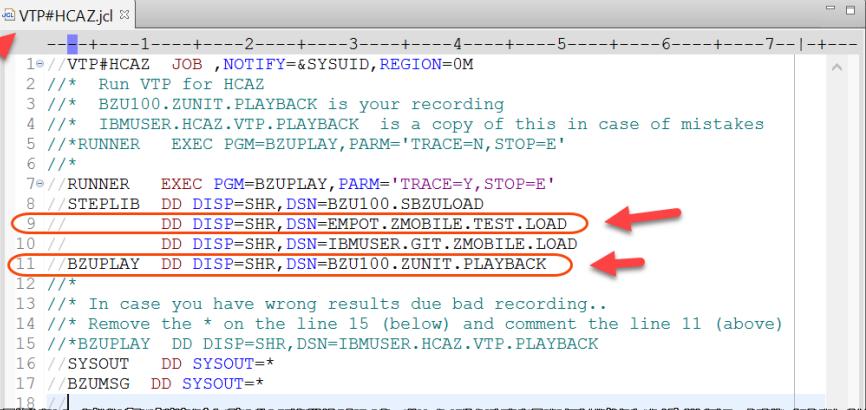
Click **OK** to connect to z/OS.



2.2.4 ► Using Z/OS Projects view (on left) expand the project **DemoHealthCare** expand **jcl** and double click on **VTP#HCAZ.jcl** to edit the JCL that will be submitted for execution.

This JCL runs a VTP test indicated by the file **BZU100.ZUNIT.PLAYBACK** tagged to the DDNAME **BZUPLAY**.

The programs executed on this test are loaded on the **EMPOT.ZMOBILE.TEST.LOAD** loadlib

```

z/OS Projects ☐
  jcl
    HCAZPLA2.jcl
    HCAZPLA3.jcl
    HCAZPLAY.jcl
    HCAZPLCC.jcl
    HCIPCC.jcl
    HCIPDBG.jcl
    HCIPZRUN.jcl
    VTP#HCAZ.jcl
    VTP#TES1.jcl
    VTP#TES2.jcl
    ZUDBDB01.jcl
    ZUDBDB02.jcl
    ZUNITCC.jcl

Properties ☐ Outline ☐

```

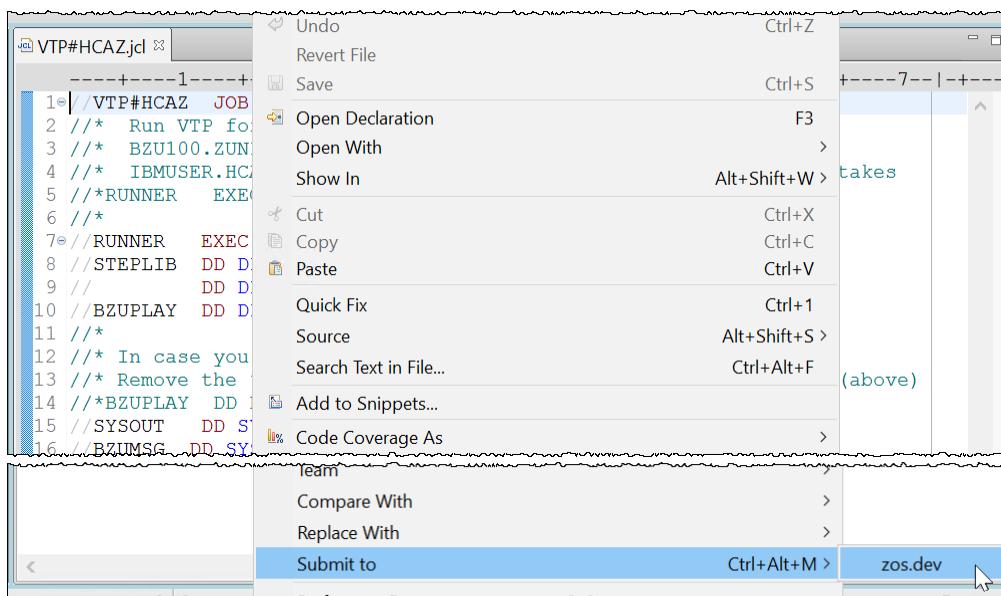
```

VTP#HCAZ.jcl ☐
-----+---1---+---2---+---3---+---4---+---5---+---6---+---7---|+-
1 //> VTP#HCAZ JOB ,NOTIFY=&SYSUID,REGION=0M
2 /* Run VTP for HCAZ
3 /* BZU100.ZUNIT.PLAYBACK is your recording
4 /* IBMUSER.HCAZ.VTP.PLAYBACK is a copy of this in case of mistakes
5 /*>RUNNER EXEC PGM=BZUPLAY,PARM='TRACE=N,STOP=E'
6 /*
7 //>RUNNER EXEC PGM=BZUPLAY,PARM='TRACE=Y,STOP=E'
8 //>STEPLIB DD DISP=SHR,DSN=BZU100.SBZULOAD
9 // DD DISP=SHR,DSN=EMPOT.ZMOBILE.TEST.LOAD
10 // DD DISP=SHR,DSN=IBMUSER.GIT.ZMOBILE.LOAD
11 // BZUPLAY DD DISP=SHR,DSN=BZU100.ZUNIT.PLAYBACK
12 /*
13 /* In case you have wrong results due bad recording..
14 /* Remove the * on the line 15 (below) and comment the line 11 (above)
15 /*>BZUPLAY DD DISP=SHR,DSN=IBMUSER.HCAZ.VTP.PLAYBACK
16 //>SYSOUT DD SYSOUT=*
17 //>BZUMSG DD SYSOUT=*
18 //

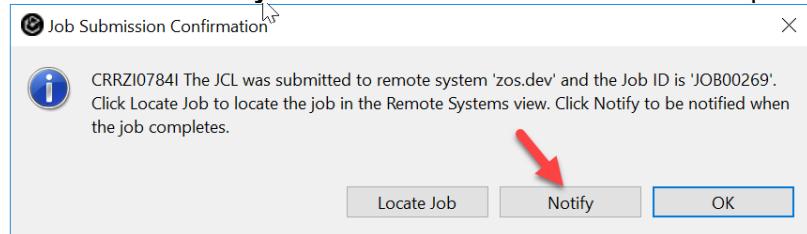
```

2.2.5 To submit this job to z/OS execution:

► Right click on the JCL being edited and select **Submit to > zos.dev** for execution on z/OS



2.2.6 ► Click **Notify** to be notified when the execution is complete.

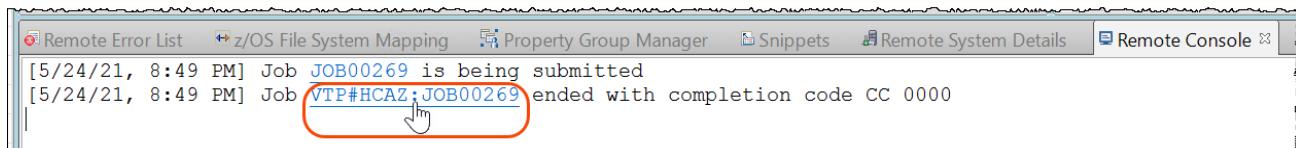


2.2.7 Under *Remote Console*, you will be notified when execution is completed.

The completion code must be 4.

► Once the execution ends, click on the link **VTP#HCAZ:JOB00xxx**

(where 00xxx is a number) to point to the Job output. This number vary depending on z/OS.



2.2.8 ► Under *Remote Systems* view expand **VTP#HCAZ:JOB00xxx** and double click **RUNNER:BZUMSG** step.

2.2.9 ► Scroll down the report displayed. In the line 28 you will see that the program **HCP1DB01** has **DB2 CALL=SELECT** and is invoked by program **HCP1BI01**

2.2.10 Scroll down to the bottom and you will see that the execution finished with RC=00.

That means that all data captured on the recording matches the execution.

Later you will introduce a bug and verify that the return code will not be 00.

```

IBMUSER.VTP#HCAZ:JOB00269.D0000102.spool
-----+---1---+---2---+---3---+---4---+---5---+---6---+---7---|+-
96 BZUP401I RECEIVED CICS CALL=X'1802'-RECEIVE_MAP PROGRAM=HCT1PL01 LINE=653
97 BZUP402I RECORDED CICS CALL=X'1802'-RECEIVE_MAP RECORD=89 ORIGINAL LINE=65
98 BZUP401I RECEIVED CICS CALL=X'1806'-SEND_TEXT PROGRAM=HCT1PL01 LINE=734
99 BZUP402I RECORDED CICS CALL=X'1806'-SEND_TEXT RECORD=91 ORIGINAL LINE=734
100 BZUP401I RECEIVED CICS CALL=X'0E08'-RETURN PROGRAM=HCT1PL01 LINE=740
101 BZUP402I RECORDED CICS CALL=X'0E08'-RETURN RECORD=92 ORIGINAL LINE=740
102 ****
103 BZUP002I FINISHED EXECUTION RC=00
104 ****
105 BZUP300I DB2 STATISTICS
106 SELECT          NORMAL=3
107 ****
108 BZUP300I CICS STATISTICS
109 HANDLE_CONDITION NORMAL=6
110 HANDLE_AID      NORMAL=6
111 LINK            NORMAL=6
112 RETURN          NORMAL=12
113 RECEIVE_MAP     NORMAL=6
114 SEND_MAP        NORMAL=6
115 SEND_TEXT       NORMAL=1
116 ****

```

2.2.11 Use **Ctrl + Shift + F4** to close all opened editors.

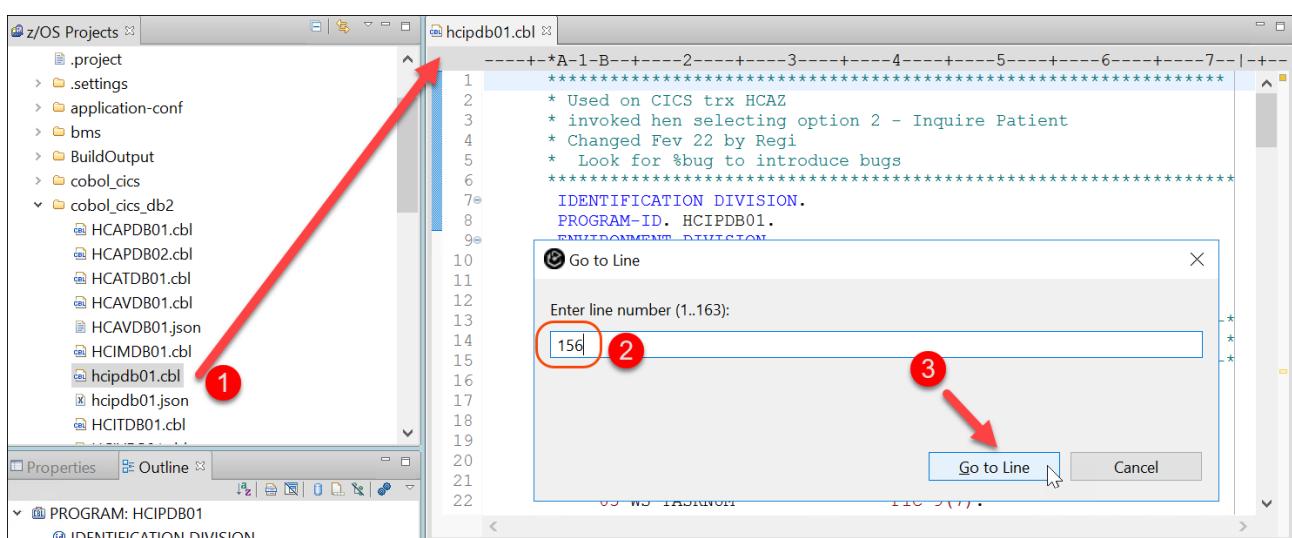
## Section 3 – Modify one program (introduce a bug) and rerun the VTP JCL

Using IDz you will modify the program and introduce a bug. Then you will rerun the batch test created with VTP in the previous sections.

### 3.1 Modifying one COBOL program introducing a bug

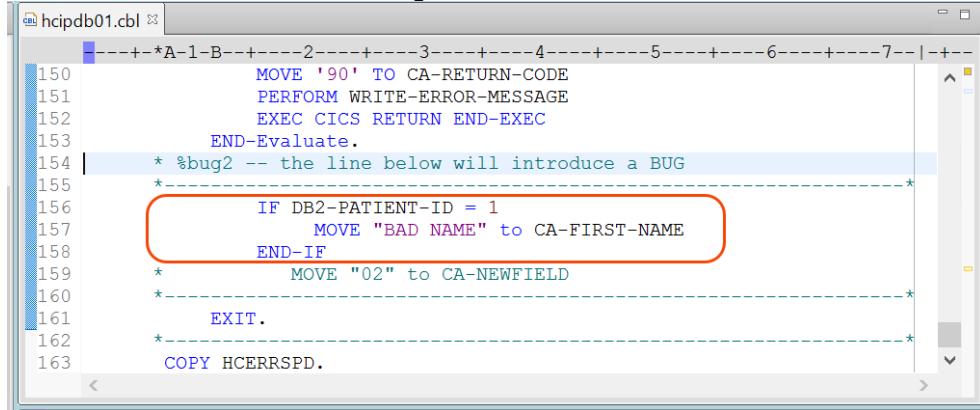
3.1.1 Using the z/OS Projects view, open **hcipdb01.cbl** under **cobol\_cics\_db2** by double clicking on it

3.1.2 On the editor, go to line **156** by pressing **Ctrl+L** typing **156** and clicking **Go to Line**.



3.1.3 ►| Change the lines 156, 157 and 158 removing the \* from the statements that move “BAD NAME”, to a COBOL field named **CA-FIRST-NAME** Tip -> Could use **Source > Toggle Comment**  
**IMPORTANT → DO NOT modify the line 159.**

►| Press **Ctrl+S** to save the changes. and **Ctrl + Shift + F4** to close all editors.



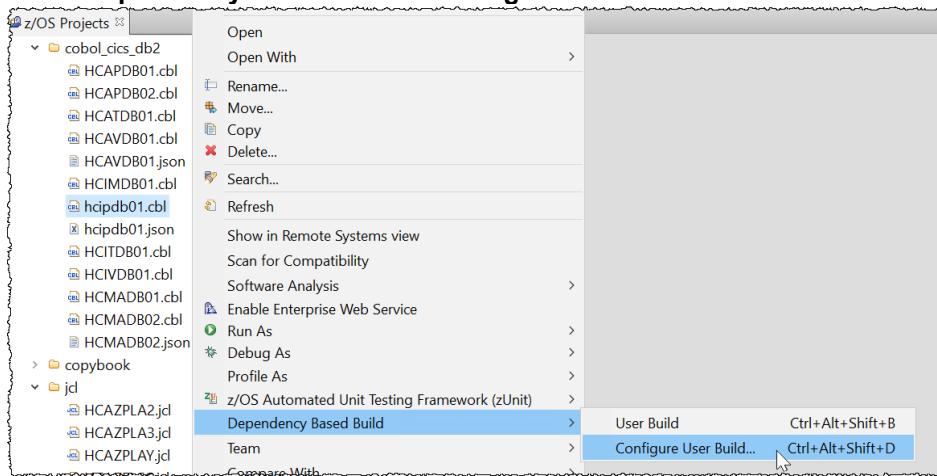
```

150      MOVE '90' TO CA-RETURN-CODE
151      PERFORM WRITE-ERROR-MESSAGE
152      EXEC CICS RETURN END-EXEC
153      END-Evaluate.
154      * %bug2 -- the line below will introduce a BUG
155      *
156      /* IF DB2-PATIENT-ID = 1
157      MOVE "BAD NAME" to CA-FIRST-NAME
158      END-IF */
159      * MOVE "02" to CA-NEWFIELD
160      *
161      EXIT.
162      *
163      COPY HCERRSPD.

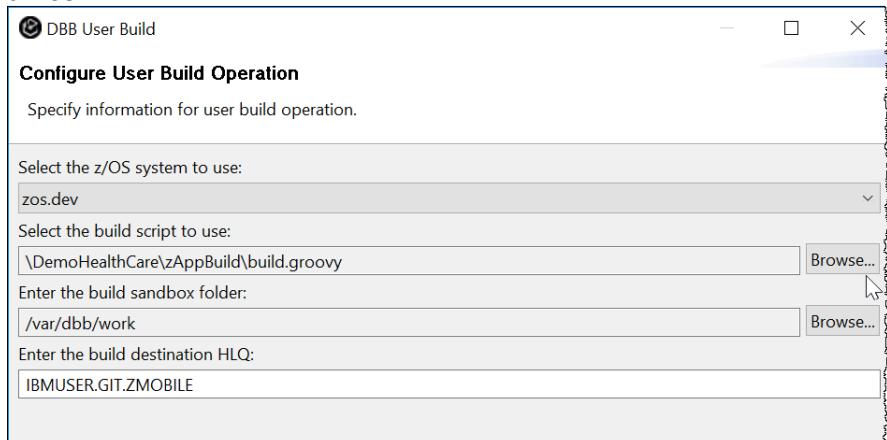
```

## 3.2 Building the modified program using DBB

3.2.1 ►| On the z/OS Projects view, right click on **hcpdb01.cbl** and select **Dependency Based Build > Configure User Build...**



3.2.2 ►| If the values are not populated, use the Browse button, specify the values below and **click Next three times**



DBB User Build

Configure User Build Operation

Specify information for user build operation.

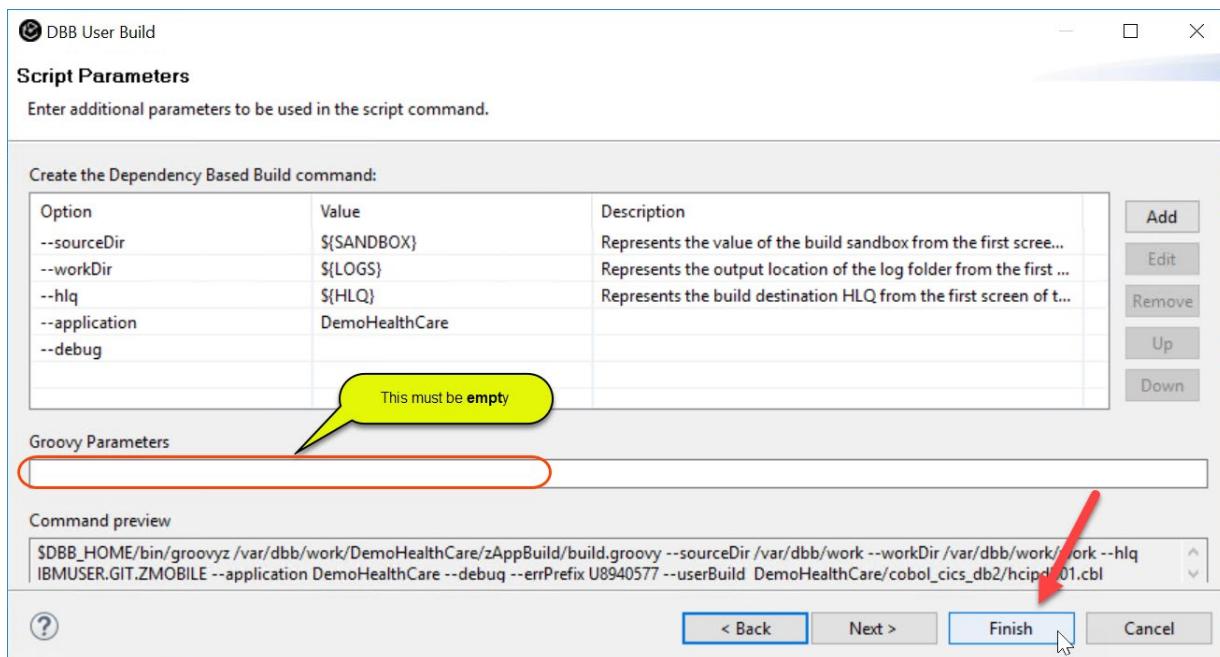
Select the z/OS system to use:  
zos.dev

Select the build script to use:  
\DemoHealthCare\zAppBuild\build.groovy

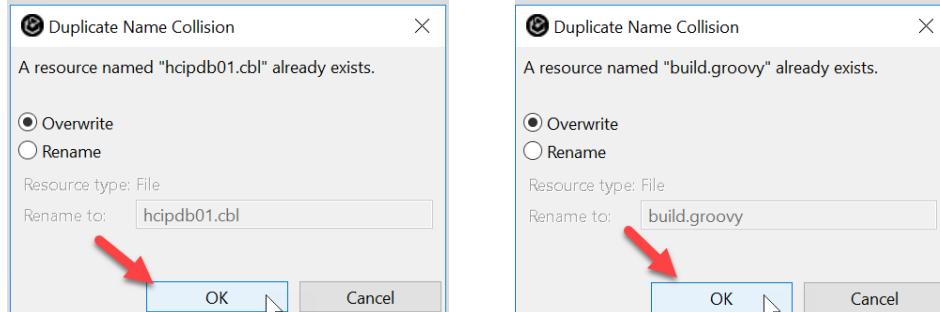
Enter the build sandbox folder:  
/var/dbb/work

Enter the build destination HLQ:  
IBMUSER.GIT.ZMOBILE

3.2.3 ➡ Be sure that the field **Groovy Parameters** is empty and click **Finish**

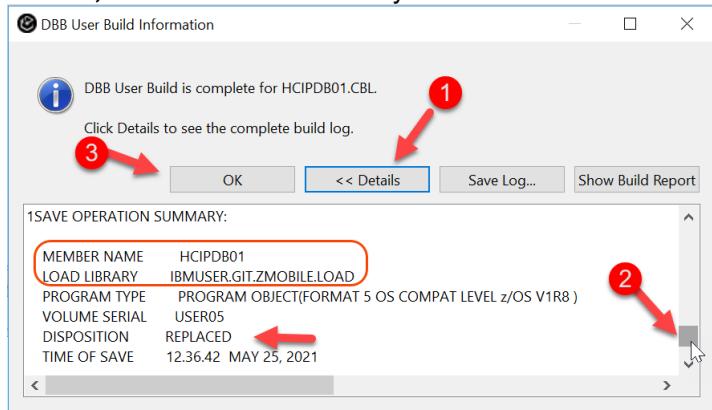


3.2.4 ➡ If asked, overwrite the code already on z/OS as the example below

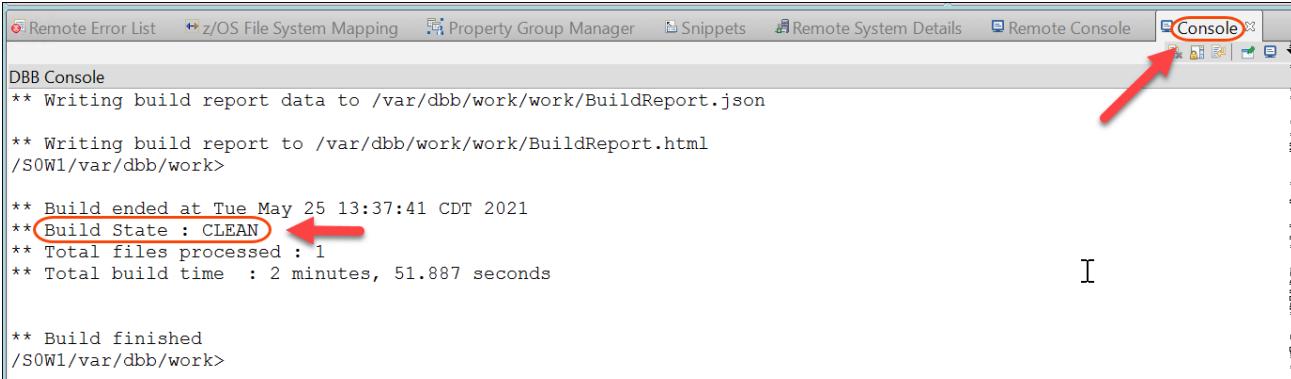


3.2.5 ➡ Click on **Console** tab (left of Remote Console) to verify what is going on..

➡ This operation may take 2 to 3 minutes. When finished the dialog below will be displayed and you can click on **Details**, scroll forward and verify that the load module was created. Then click **OK**.



3.2.6 ➡ Click on the **Console** view to see the results:



```

DBB Console
** Writing build report data to /var/dbb/work/work/BuildReport.json

** Writing build report to /var/dbb/work/work/BuildReport.html
/SOW1/var/dbb/work

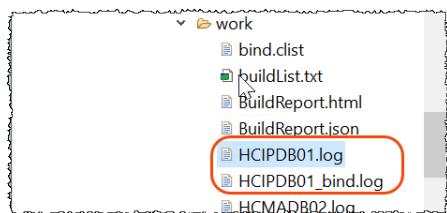
** Build ended at Tue May 25 13:37:41 CDT 2021
** Build State : CLEAN
** Total files processed : 1
** Total build time : 2 minutes, 51.887 seconds

** Build finished
/SOW1/var/dbb/work

```

3.2.7 The logs of the COBOL Compiler/Link is at **z/OS UNIX Files** on  
**/var/dbb/work/work/HCIPDB01.log**

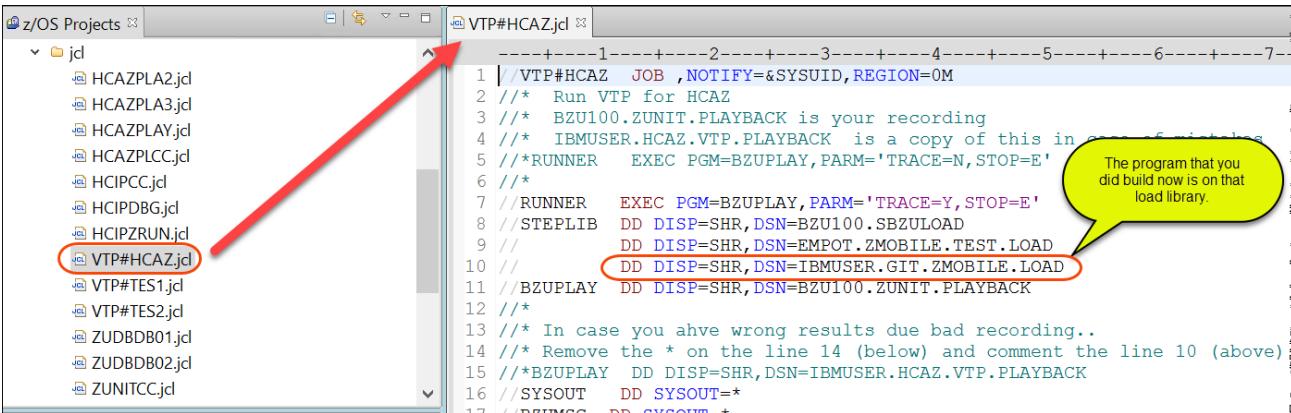
The log of the DB2 Bind is at **/var/dbb/work/work/HCIPDB01\_bind.log**



### 3.3 Running the VTP JCL again against the modified program

Since we introduced a bug, now the VTP test case should fail.

3.3.1 ➡ Using the project **DemoHealthCare** expand **jcl** and double click on **VTP#HCAZ.jcl** to edit the JCL that will invoke the VTP.



```

VTP#HCAZ.jcl
-----+---1---+---2---+---3---+---4---+---5---+---6---+---7
1 //VTP#HCAZ JOB ,NOTIFY=&SYSUID,REGION=0M
2 //* Run VTP for HCAZ
3 // BZU100.ZUNIT.PLAYBACK is your recording
4 // IBMUSER.HCAZ.VTP.PLAYBACK is a copy of this in case you ahve wrong results due bad recording..
5 //RUNNER EXEC PGM=BZUPLAY,PARM='TRACE=N,STOP=E'
6 /**
7 //RUNNER EXEC PGM=BZUPLAY,PARM='TRACE=Y,STOP=E'
8 //STEPLIB DD DISP=SHR,DSN=BZU100.SBZULOAD
9 //          DD DISP=SHR,DSN=EMPOT.ZMOBILE.TEST.LOAD
10 //         DD DISP=SHR,DSN=IBMUSER.GIT.ZMOBILE.LOAD
11 //BZUPLAY DD DISP=SHR,DSN=BZU100.ZUNIT.PLAYBACK
12 /**
13 /* In case you ahve wrong results due bad recording..
14 /* Remove the * on the line 14 (below) and comment the line 10 (above)
15 //BZUPLAY DD DISP=SHR,DSN=IBMUSER.HCAZ.VTP.PLAYBACK
16 //SYSOUT DD SYSOUT=*
17 BZUMSG DD SYSOUT=*

```

### 3.3.2 The modified program is now at **IBMUSER.GIT.ZMOBILE.LOAD**

► Add an \* as below to comment the PDS **EMPOT.ZMOBILE.TEST.LOAD**

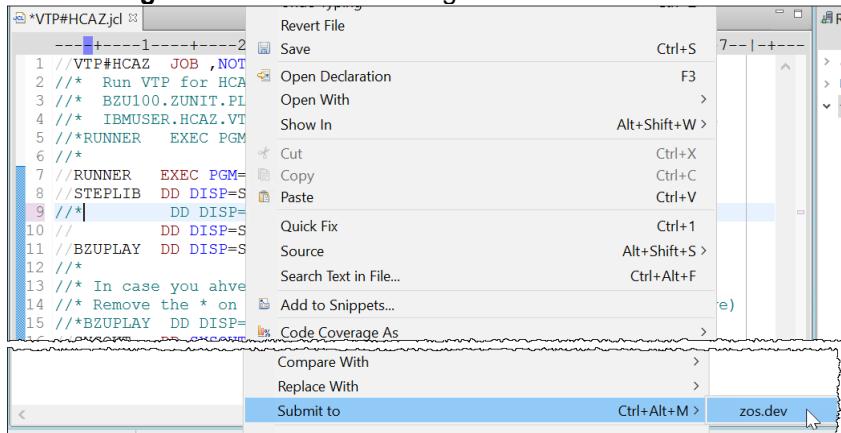
```

1 //> /VTP#HCAZ JOB ,NOTIFY=&SYSUID,REGION=0M
2 //> Run VTP for HCAZ
3 ///* BZU100.ZUNIT.PLAYBACK is your recording
4 ///* IBMUSER.HCAZ.VTP.PLAYBACK is a copy of this in case of mistakes
5 ///*RUNNER EXEC PGM=BZUPLAY,PARM='TRACE=N,STOP=E'
6 ///*
7 //RUNNER EXEC PGM=BZUPLAY,PARM='TRACE=Y,STOP=E'
8 //STEPLIB DD DISP=SHR,DSN=BZU100.SBZULOAD
9 ///*| DD DISP=SHR,DSN=EMPOT.ZMOBILE.TEST.LOAD
10 // DD DISP=SHR,DSN=IBMUSER.GIT.ZMOBILE.LOAD
11 //BZUPLAY DD DISP=SHR,DSN=BZU100.ZUNIT.PLAYBACK
12 ///*
13 ///* In case you ahve wrong results due bad recording..
14 ///* Remove the * on the line 14 (below) and comment the line 10 (above)
15 //BZUPLAY DD DISP=SHR,DSN=IBMUSER.HCAZ.VTP.PLAYBACK
16 //SYSOUT DD SYSOUT=*
17 //BZUMSG DD SYSOUT=*
18 //

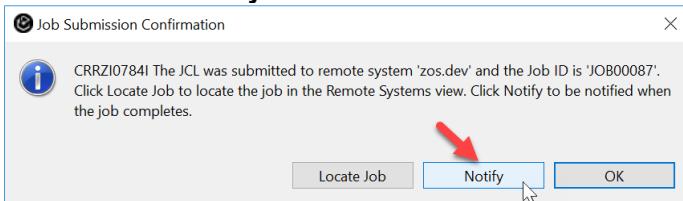
```

### 3.3.3 ► Use **Ctrl + S** to save the JCL updates,

### 3.3.4 ► Right click on the JCL being edited and select **Submit to > zos.dev** for execution on z/OS



### 3.3.5 ► Click **Notify** to be notified when the execution is complete.

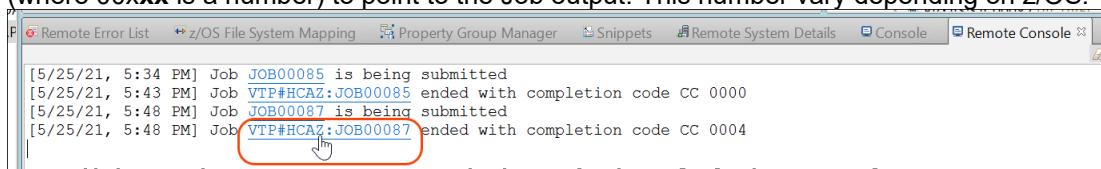


### 3.3.6 Under **Remote Console**, you will be notified when execution is completed.

The completion code must be 4.

► Once the execution ends, **click on the link** **VTP#HCAZ:JOB00xxx**

(where 00xxx is a number) to point to the Job output. This number vary depending on z/OS.



3.3.7 ► Under Remote Systems view expand **VTP#HCAZ:JOB00xxx (CC0004)** and double click **RUNNER:BZUMSG** step.

3.3.8 ► Scroll down the report displayed and on line 35 you will see that the program **HCIPDB01** has **BAD NAME** while the recorded data was **Ralph**

```

20 BZUP401I RECEIVED CICS CALL=X'0E02'-LINK PROGRAM=HCIPDB01 LINE=220
21 BZUP402I RECORDED CICS CALL=X'0E02'-LINK RECORD=22 ORIGINAL LINE=220
22 BZUP7001 RECEIVED DB2 CALL=SELECT PROGRAM=HCIPDB01 LINE=235
23 BZUP701I RECORDED DB2 CALL=SELECT RECORD=25 ORIGINAL LINE=247
24 BZUP401I RECEIVED CICS CALL=X'0E08'-RETURN PROGRAM=HCIPDB01 LINE=230
25 BZUP402I RECORDED CICS CALL=X'0E08'-RETURN RECORD=26 ORIGINAL LINE=241
26 BZUP202W REQUEST=RETURN ARG=CMA RECORD=26 ADDRESS=20757100 OFFSET=28
27 PROGRAM=HCIPDB01 LINE=230 MODE=INPT
28 PROGRAM COLS=-+---3---+---4---+---5---+---6---+---7---+
29 PROGRAM DATA=1234BAD NAME DALmeida 1980-07-1134 Main
30 PROGRAM DATA=FFFFFCCC4DCDC44CC9988844444444444FFFF6FF6FFFF4D889
31 PROGRAM DATA=12342140514500413459410000000000019800070113404195
32 BZUP202W REQUEST=RETURN ARG=CMA RECORD=26 ADDRESS=20757100 OFFSET=28
33 PROGRAM=HCIPDB01 LINE=230 MODE=INPT
34 PROGRAM COLS=-+---3---+---4---+---5---+---6---+---7---+
35 PROGRAM DATA=1234BAD NAME DALmeida 1980-07-1134 Main
36 PROGRAM DATA=FFFFFCCC4DCDC44CC9988844444444444FFFF6FF6FFFF4D889
37 PROGRAM DATA=12342140514500413459410000000000019800070113404195
38 RECORDED DATA=1234Ralph DALmeida 1980-07-1134 Main
39 RECORDED DATA=FFFFFD899844444CC9988844444444444FFFF6FF6FFFF4D889
40 RECORDED DATA=1234913780000413459410000000000019800070113404195
41 BZUP401I RECEIVED CICS CALL=X'0E08'-RETURN PROGRAM=HCIPDB01 LINE=212
42 BZUP402I RECORDED CICS CALL=X'0E08'-RETURN RECORD=28 ORIGINAL LINE=212
43 BZUP202W REQUEST=RETURN ARG=CMA RECORD=28 ADDRESS=20757100 OFFSET=28
44 PROGRAM=HCIPDB01 LINE=212 MODE=INPT
45 PROGRAM COLS=-+---3---+---4---+---5---+---6---+---7---+
46 PROGRAM DATA=1234BAD NAME DALmeida 1980-07-1134 Main
47 PROGRAM DATA=FFFFFCCC4DCDC44CC9988844444444444FFFF6FF6FFFF4D889
48 PROGRAM DATA=12342140514500413459410000000000019800070113404195

```

3.3.9 This means that the data returned by the program HCIPDB01 is not the same as it was recorded. That could indicate a fail on the program.

► Scroll down to the bottom and you will see that the execution finished with RC=04. You will see various mismatches. That means that not all data captured on the recording matched during the execution.

```

213 BZUP401I RECEIVED CICS CALL=X'1802'-RECEIVE_MAP PROGRAM=HCT1PL01 LINE=653
214 BZUP402I RECORDED CICS CALL=X'1802'-RECEIVE_MAP RECORD=89 ORIGINAL LINE=65
215 BZUP401I RECEIVED CICS CALL=X'1806'-SEND_TEXT PROGRAM=HCT1PL01 LINE=734
216 BZUP402I RECORDED CICS CALL=X'1806'-SEND_TEXT RECORD=91 ORIGINAL LINE=734
217 BZUP401I RECEIVED CICS CALL=X'0E08'-RETURN PROGRAM=HCT1PL01 LINE=740
218 BZUP402I RECORDED CICS CALL=X'0E08'-RETURN RECORD=92 ORIGINAL LINE=740
219 ****
220 BZUP002I FINISHED EXECUTION RC=04
221 ****
222 BZUP300I DB2 STATISTICS
223 SELECT NORMAL=3
224 ****
225 BZUP300I CICS STATISTICS
226 HANDLE_CONDITION NORMAL=6
227 HANDLE_AID NORMAL=6
228 LINK NORMAL=6
229 RETURN NORMAL=12
230 RECEIVE_MAP NORMAL=6
231 SEND_MAP NORMAL=6
232 SEND_TEXT NORMAL=1
233 ****
234

```

3.3.10 ►| Use **Ctrl + Shift + F4** to close all opened editors.

#### What have you done so far?

**On Section 1** -You executed the CICS transaction **HCAZ** and using VTP you recorded a simple interaction with the *Health Care application*. The recorded data is saved on a Z/OS dataset..



**On Section 2** -You executed the VTP JCL that execute the sequence recorded. Since you made no changes the return code must be 00.

**On Section 3** -You modified the COBOL program and introduced a bug. The program modified was compiled and DB2 was bind using the DBB User Build capability. Again the VTP JCL is submitted for batch execution and we can verify the bug on the batch output execution.



You introduced a bug on the application.

On the next part you must fix it and verify that its fixed using the **IBM Z VTP**.

### PART #2 – Fix the program bug and re-run the test

## Section 4. Run the CICS transaction and verify the bug .

Since the modified load module was deployed to a loadlib that is on the datasets that CICS uses you could see the bug on the execution. You may need to execute a CICS NEWCOPY to see that.  
Below the details

### 4.1 Issuing a CICS New copy using 3270 emulation terminal

4.1.1 ►| Bring up a 3270 terminal emulator clicking on the **host emulator icon** on the Windows task bar.



This opens the host emulator.

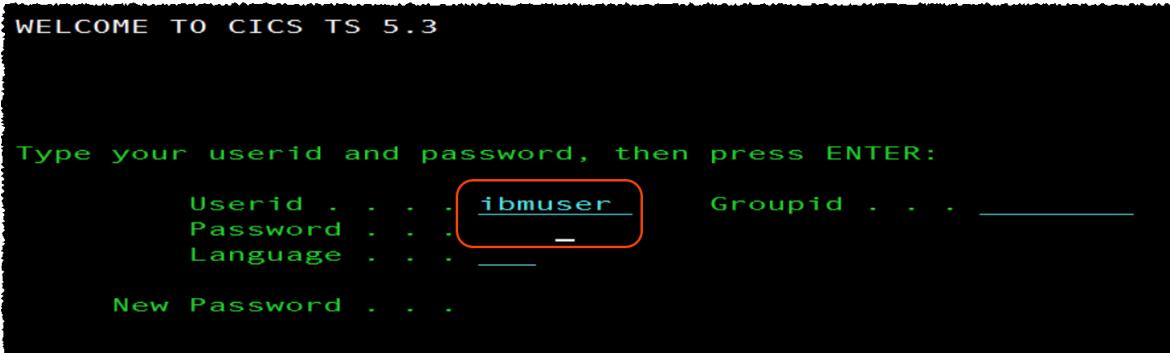
4.1.2 ►| Type **I cicsts53**. (where I is L lower case) and press **Enter key**.

```
====> Enter "LOGON" followed by the TSO userid. Example "LOGON IBMUSER" or  
====> Enter L followed by the APPLID  
====> Examples: "L TSO", "L CICSTS52", "L CICSTS53", "L IMS13", "L IMS14"
```

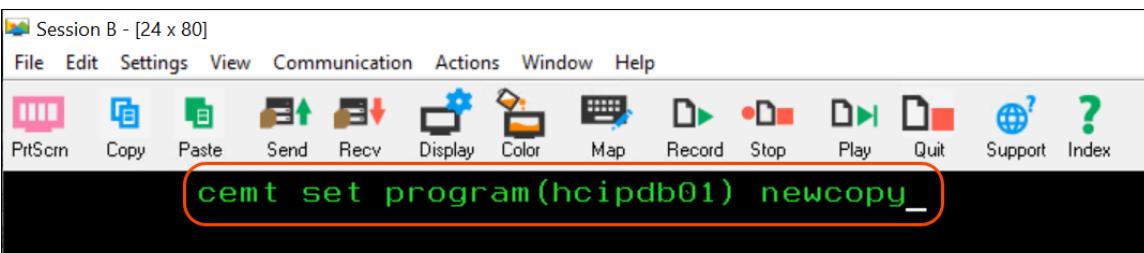
```
I cicsts53
```

24/01/11

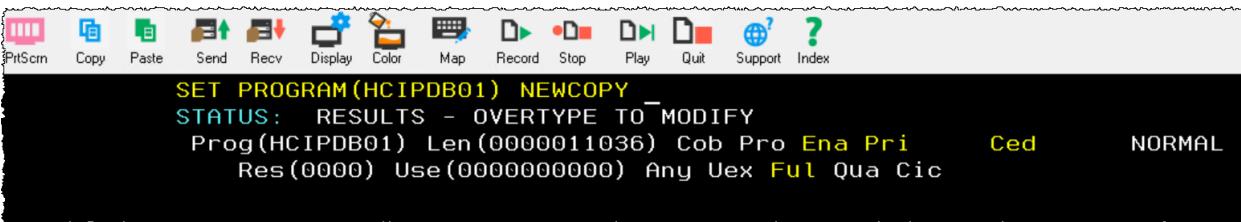
4.1.3 ► Sign on using **ibmuser** as the userid and **sys1** as the password and press **Enter key**.



4.1.4 ► Type **cemt set program(hcipdb01) newcopy** and press **Enter key**.



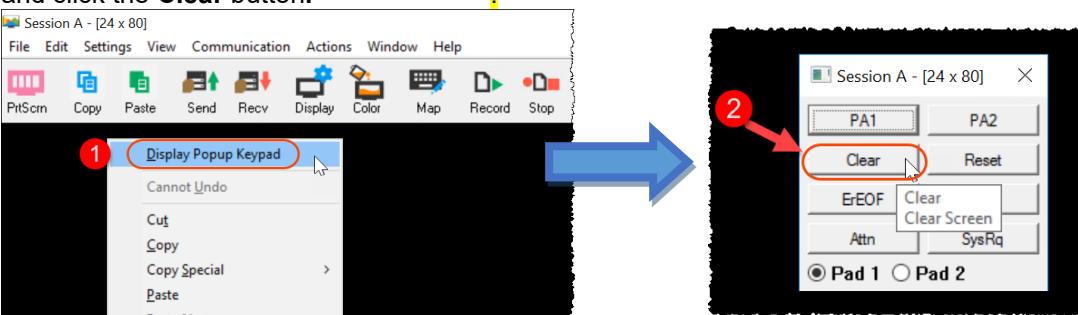
4.1.5 This below indicates the **NEWCOPY** was successful:



4.1.6 ► Press **PF3** to end the dialog

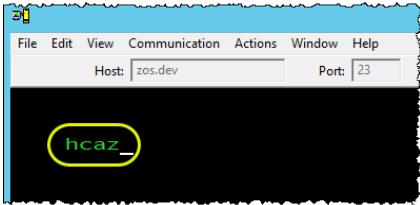


4.1.7 ► To clear the screen, **right click** on the dark space and select **Display Popup Keypad** and click the **Clear** button.

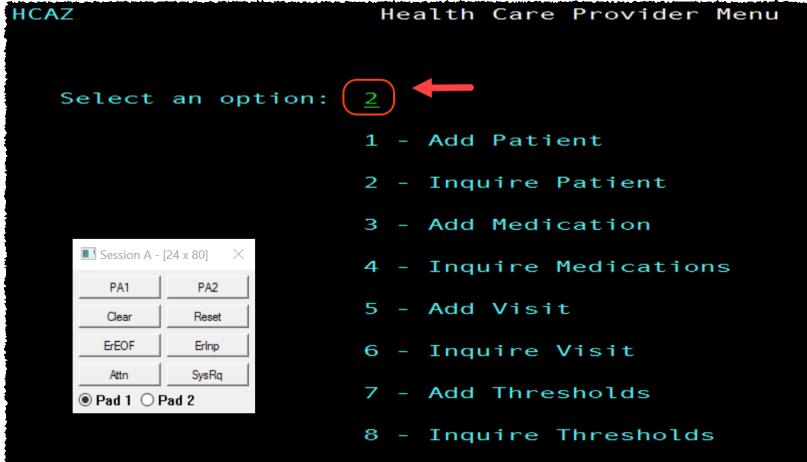


## 4.2 Executing HCAZ transaction

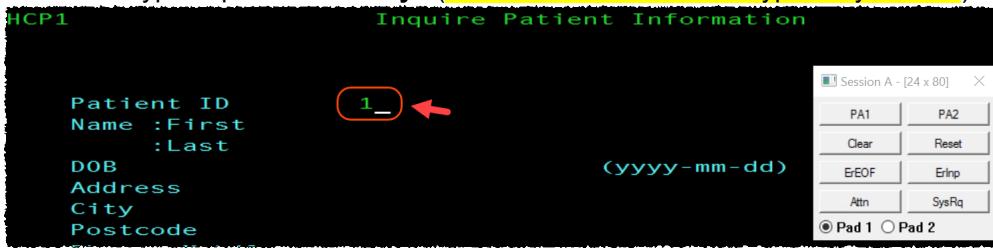
4.2.1 ► On CICS main terminal type **hcaz** and press **Enter key** to invoke the CICS transaction application



4.2.2 ► Type 2 press **Enter key**.



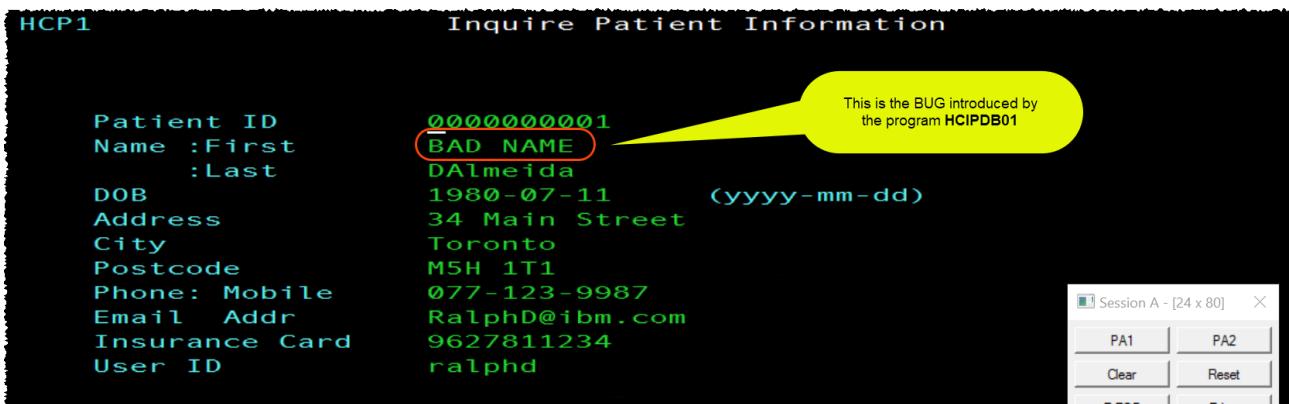
4.2.3 ► Type 1 press **Enter key**. (You can close the stick keypad if you want)



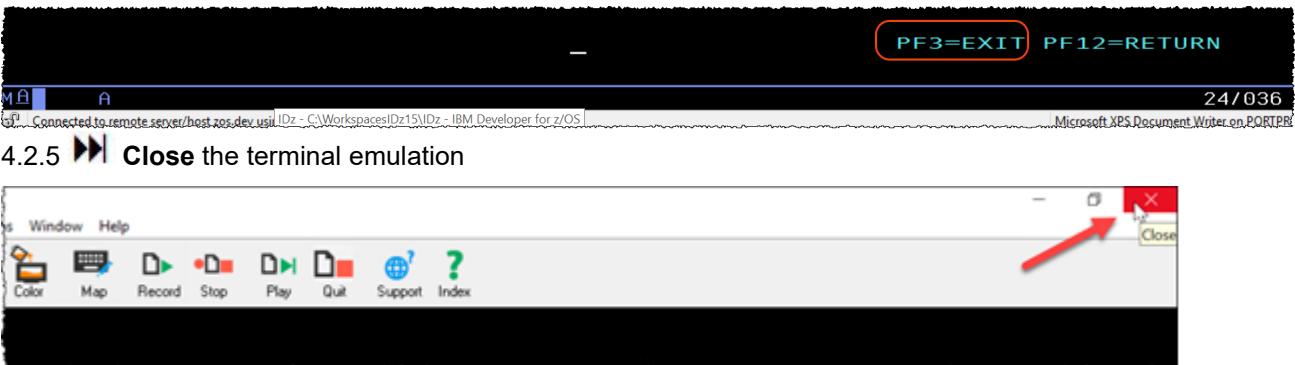
You should have the patient data displayed.

Notice that instead of **Ralph** the **BAD NAME** is displayed

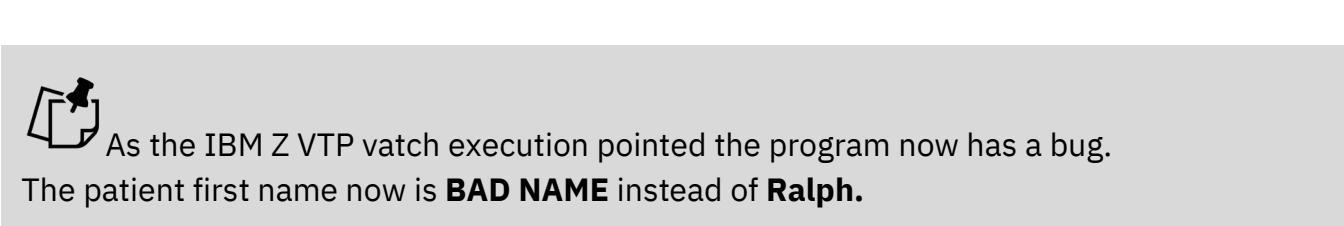
This confirm the report provided by the batch JCL executed before.



4.2.4 ➡ Press PF3 to exit the program



4.2.5 ➡ Close the terminal emulation



As the IBM Z VTP vatch execution pointed the program now has a bug.  
The patient first name now is **BAD NAME** instead of **Ralph**.

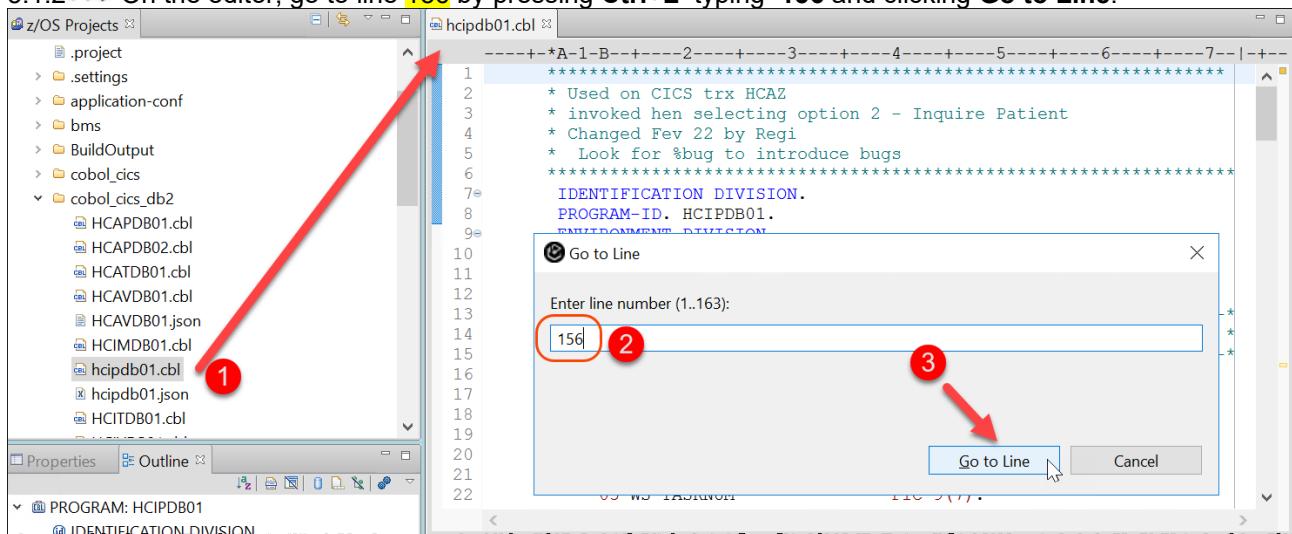
## Section 5 Use IDz to fix the bug and recompile/bind the program.

Using IDz again you will remove the bug introduced Below the details

### 5.1 Modifying one COBOL program again to remove the bug

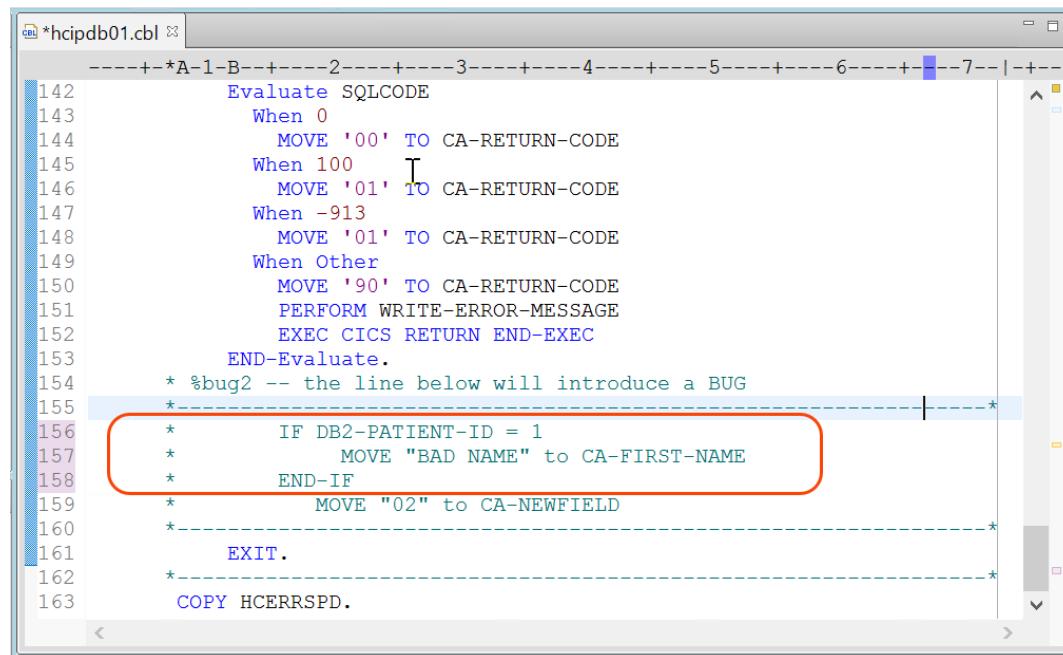
5.1.1 ➡ Using IDz and the z/OS Projects view, open **hcipdb01.cbl** under **cobol\_cics\_db2** by double clicking on it

5.1.2 ➡ On the editor, go to line 156 by pressing **Ctrl+L** typing **156** and clicking **Go to Line**.



5.1.3 ►| Change the lines 156, 157 and 158 adding the \* on column 7 to the statements that move "BAD NAME", to a COBOL field named CA-FIRST-NAME **Tip -> Could use Source > Toggle Comment**

►| Press **Ctrl+S** to save the changes. and **Ctrl + Shift + F4** to close all editors.



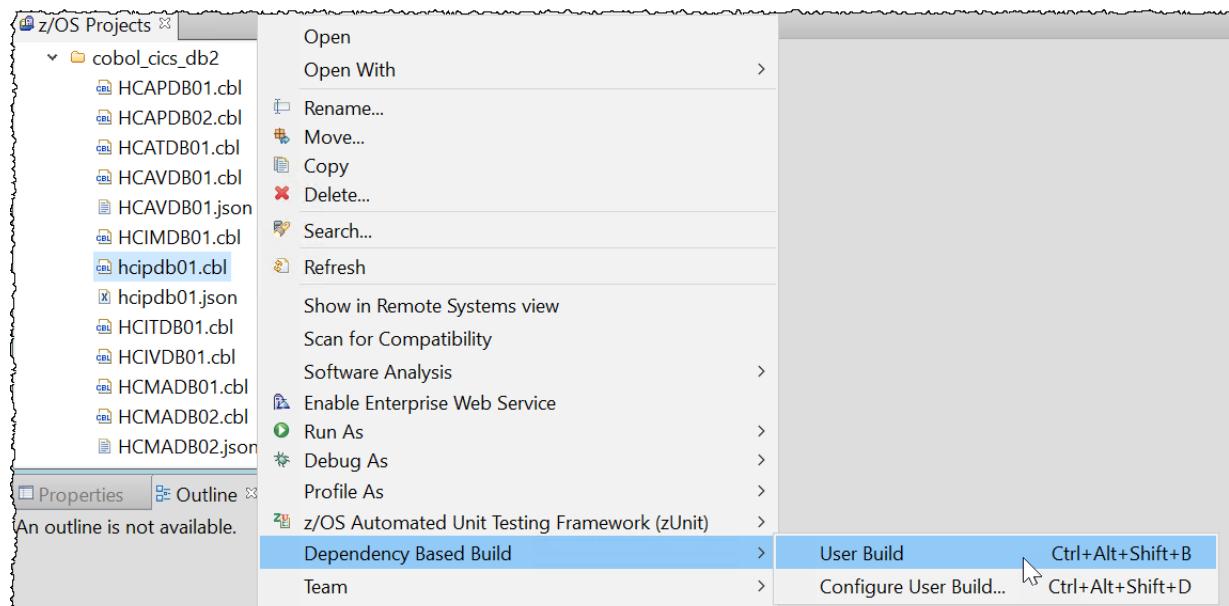
```

142      Evaluate SQLCODE
143      When 0
144          MOVE '00' TO CA-RETURN-CODE
145      When 100
146          MOVE '01' TO CA-RETURN-CODE
147      When -913
148          MOVE '01' TO CA-RETURN-CODE
149      When Other
150          MOVE '90' TO CA-RETURN-CODE
151          PERFORM WRITE-ERROR-MESSAGE
152          EXEC CICS RETURN END-EXEC
153      END-Evaluate.
154      * %bug2 -- the line below will introduce a BUG
155      *
156      *     IF DB2-PATIENT-ID = 1
157      *         MOVE "BAD NAME" to CA-FIRST-NAME
158      *     END-IF
159      *     MOVE "02" to CA-NEWFIELD
160      *
161      EXIT.
162      *
163      COPY HCERRSPD.

```

## 5.2 Rebuilding the fixed program using DBB

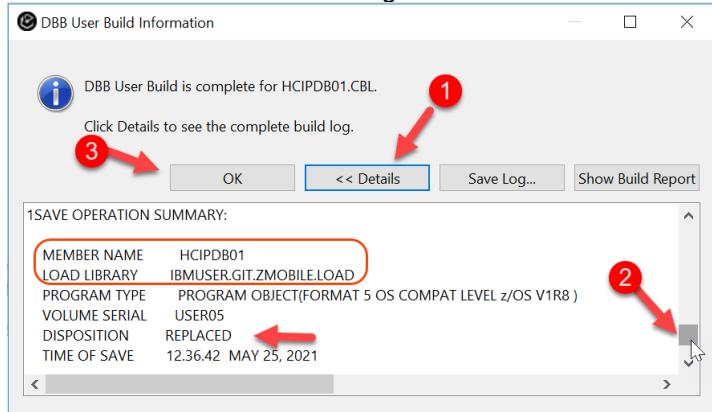
5.2.1 ►| On the z/OS Projects view, right click on **hcipdb01.cbl** and select **Dependency Based Build > User Build...**



5.2.3 ► Click on the **Console** view (left of Remote Console) to see the results.

This operation may take up to 2 minutes

► When finished click on **Details**, advance forward and verify that the load module was created. Then click **OK** to close the dialog.



5.2.4 ► Click on the **Console** view to see the results:

```

Remote Error List  /OS File System Mapping  Property Group Manager  Snippets  Remote System Details  Remote Console  Console  Help  Log  Refresh  Minimize  Maximize  Close

DBB Console
** Writing build report data to /var/dbb/work/work/BuildReport.json
** Writing build report to /var/dbb/work/work/BuildReport.html
/SOW1/var/dbb/work

** Build ended at Tue May 25 13:37:41 CDT 2021
** Build State : CLEAN
** Total files processed : 1
** Total build time : 2 minutes, 51.087 seconds

** Build finished
/SOW1/var/dbb/work>

```

The logs of the COBOL Compiler/Link is at </var/dbb/work/work/HCIPDB01.log>

The log of the DB2 Bind is at [/var/dbb/work/work/HCIPDB01\\_bind.log](/var/dbb/work/work/HCIPDB01_bind.log)

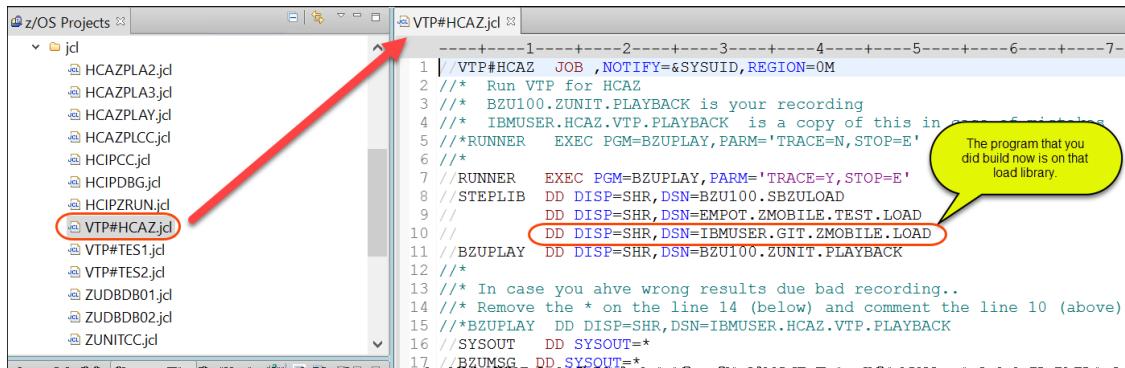
## Section 6 Rerun the VTP JCL and verify that the bug is eliminated.

Once you fixed the program you may resubmit the VTP JCL and verify, that there is no errors

### 6.1 Running the VTP JCL again against the modified program

Since we fixed the bug , now the VTP test should execute with Return Code 00..

6.1.1 ► Using the project **DemoHealthCare** expand **jcl** and double click on **VTP#HCAZ.jcl** to edit the JCL that runs VTP.



### 6.1.2 The modified program is now at **IBMUSER.GIT.ZMOBILE.LOAD**

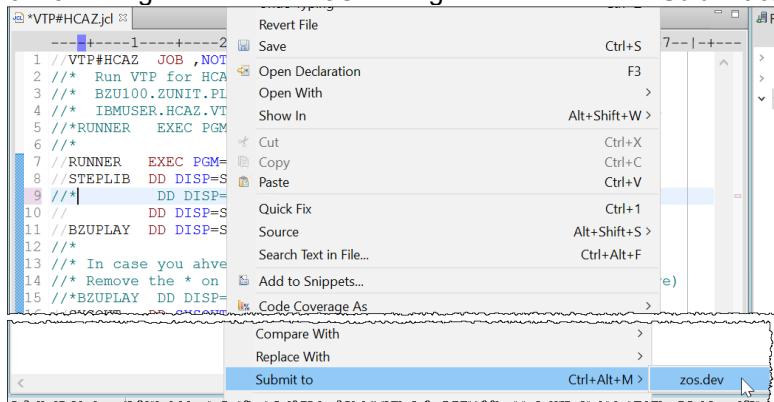
► If necessary add an \* as below to comment the PDS **EMPOT.ZMOBILE.TEST.LOAD**

```

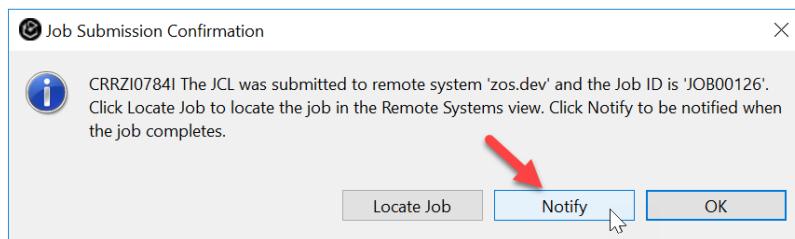
*VTP#HCAZ.jcl
1 //> VTP#HCAZ JOB ,NOTIFY=&SYSUID,REGION=0M
2 //> Run VTP for HCAZ
3 ///* BZU100.ZUNIT.PLAYBACK is your recording
4 ///* IBMUSER.HCAZ.VTP.PLAYBACK is a copy of this in case of mistakes
5 ///*RUNNER EXEC PGM=BZUPLAY,PARM='TRACE=N,STOP=E'
6 /**
7 ///*RUNNER EXEC PGM=BZUPLAY,PARM='TRACE=Y,STOP=E'
8 //STEPLIB DD DISP=SHR,DSN=BZU100.SBZULOAD
9 ///*| DD DISP=SHR,DSN=EMPOT.ZMOBILE.TEST.LOAD
10 //| DD DISP=SHR,DSN=IBMUSER.GIT.ZMOBILE.LOAD
11 //BZUPLAY DD DISP=SHR,DSN=BZU100.ZUNIT.PLAYBACK
12 /**
13 ///* In case you ahve wrong results due bad recording..
14 ///* Remove the * on the line 14 (below) and comment the line 10 (above)
15 //BZUPLAY DD DISP=SHR,DSN=IBMUSER.HCAZ.VTP.PLAYBACK
16 //SYSOUT DD SYSOUT=*
17 //BZUMSG DD SYSOUT=*
18 //

```

### 6.1.3 ► Right click on the JCL being edited and select **Submit to > zos.dev** for execution on z/OS



### 6.1.4 ► Click **Notify** to be notified when the execution is complete.

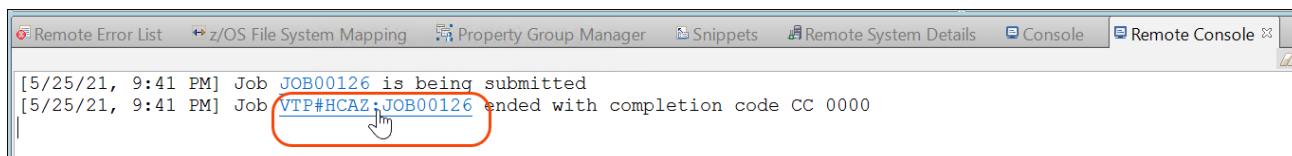


### 6.1.5 Under **Remote Console**, you will be notified when execution is completed.

**The completion code must be 00.**

► Once the execution ends, **click on the link VTP#HCAZ:JOB00xxx**

(where 00xxx is a number) to point to the Job output. This number vary depending on z/OS.



6.1..6 ► Under Remote Systems view expand **VTP#HCAZ:JOB00xxx** and double click **RUNNER:BZUMSG** step.

6.1.7 ► Scroll down the report displayed and you will see the program HCIPDB01 returned data matches what you have recorded originally. **No errors any longer**

The screenshot shows the Rational Developer for System z interface. On the left is a terminal window titled 'IBMUSER.VTP#HCAZ.JOB00126.D0000102.?spool' displaying a log of CICS transactions. A red box highlights the lines from 28 to 32, which correspond to the recorded steps in the Remote Systems tree view on the right. The tree view shows a hierarchy under 'JES' / 'Retrieved Jobs' for job 'VTP#HCAZ:JOB00126'. The 'RUNNER:BZUMSG' step is highlighted with a red circle. A red arrow points from the terminal window to the tree view.

```

CL *VTP#HCAZ.jcl IBMUSER.VTP#HCAZ.JOB00126.D0000102.?spool
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8-----+
22 BZUP401I RECEIVED CICS CALL=X'1802'-RECEIVE_MAP PROGRAM=HCP1PL01 LINE=658
23 BZUP402I RECORDED CICS CALL=X'1802'-RECEIVE_MAP RECORD=19 ORIGINAL LINE=65
24 BZUP401I RECEIVED CICS CALL=X'0E02'-LINK PROGRAM=HCP1PL01 LINE=669
25 BZUP402I RECORDED CICS CALL=X'0E02'-LINK RECORD=20 ORIGINAL LINE=669
26 BZUP401I RECEIVED CICS CALL=X'0E02'-LINK PROGRAM=HCP1B101 LINE=220
27 BZUP402I RECORDED CICS CALL=X'0E02'-LINK RECORD=22 ORIGINAL LINE=220
28 BZUP700I RECEIVED DB2 CALL=SELECT PROGRAM=HCIPDB01 LINE=235
29 BZUP701I RECORDED DB2 CALL=SELECT RECORD=25 ORIGINAL LINE=247
30 BZUP401I RECEIVED CICS CALL=X'0E08'-RETURN PROGRAM=HCIPDB01 LINE=230
31 BZUP402I RECORDED CICS CALL=X'0E08'-RETURN RECORD=26 ORIGINAL LINE=241
32 BZUP401I RECEIVED CICS CALL=X'0E08'-RETURN PROGRAM=HCP1B101 LINE=212
33 BZUP402I RECORDED CICS CALL=X'0E08'-RETURN RECORD=28 ORIGINAL LINE=212
34 BZUP401I RECEIVED CICS CALL=X'1804'-SEND_MAP PROGRAM=HCP1PL01 LINE=688
35 BZUP402I RECORDED CICS CALL=X'1804'-SEND_MAP RECORD=31 ORIGINAL LINE=688
36 BZUP401I RECEIVED CICS CALL=X'0E08'-RETURN PROGRAM=HCP1PL01 LINE=747
37 BZUP402I RECORDED CICS CALL=X'0E08'-RETURN RECORD=32 ORIGINAL LINE=747
38 BZUP400I STARTING TRANSACTION=HCP1 USING PROGRAM=HCP1PL01
39 BZUP401I RECEIVED CICS CALL=X'0206'-HANDLE_AID PROGRAM=HCP1PL01 LINE=649
40 RZUP402I RECORDED CICS CALL=X'0206'-HANDLE_AID RECORD=35 ORTGPNAT, T_TNRE=649

```

6.1.8 ► Scroll down to the bottom and you will see that the execution finished with RC=00. That means that all data captured on the recording matches the execution.

The screenshot shows the Rational Developer for System z interface. On the left is a terminal window titled 'IBMUSER.VTP#HCAZ.JOB00269.D0000102.?spool' displaying a log of CICS transactions. A red box highlights the line 'BZUP002I FINISHED EXECUTION RC=00'. A red arrow points from this line to the bottom of the terminal window, where a cursor is visible. The tree view on the right shows a hierarchy under 'JES' / 'Retrieved Jobs' for job 'VTP#HCAZ:JOB00269'. The 'RUNNER:BZUMSG' step is highlighted with a red circle. A red arrow points from the terminal window to the tree view.

```

CL *VTP#HCAZ.jcl IBMUSER.VTP#HCAZ.JOB00269.D0000102.?spool
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8-----+
96 BZUP401I RECEIVED CICS CALL=X'1802'-RECEIVE_MAP PROGRAM=HCT1PL01 LINE=653
97 BZUP402I RECORDED CICS CALL=X'1802'-RECEIVE_MAP RECORD=89 ORIGINAL LINE=65
98 BZUP401I RECEIVED CICS CALL=X'1806'-SEND_TEXT PROGRAM=HCT1PL01 LINE=734
99 BZUP402I RECORDED CICS CALL=X'1806'-SEND_TEXT RECORD=91 ORIGINAL LINE=734
100 BZUP401I RECEIVED CICS CALL=X'0E08'-RETURN PROGRAM=HCT1PL01 LINE=740
101 BZUP402I RECORDED CICS CALL=X'0E08'-RETURN RECORD=92 ORIGINAL LINE=740
102 ****
103 BZUP002I FINISHED EXECUTION RC=00
104 ****
105 BZUP300I DB2 STATISTICS
106 SELECT          NORMAL=3
107 ****
108 BZUP300I CICS STATISTICS
109 HANDLE_CONDITION NORMAL=6
110 HANDLE_AID      NORMAL=6
111 LINK            NORMAL=6
112 RETURN          NORMAL=12
113 RECEIVE_MAP     NORMAL=6
114 SEND_MAP        NORMAL=6
115 SEND_TEXT       NORMAL=1
116 ****

```

6.1.9 ► Use **Ctrl + Shift + F4** to close all opened editors.

Notice that this capability allows you to invoke Z VTP execution using pipelines like Jenkins.

#### What have you done so far?

**On Section 1** -You executed the CICS transaction **HCAZ** and using VTP you recorded a simple interaction with the *Health Care application*. The recorded data is saved on a Z/OS dataset..

**On Section 2** -You executed the VTP JCL that execute the sequence recorded. Since you made no changes the return code must be 00.

**On Section 3** -You modified the COBOL program and introduced a bug. The program modified was compiled and DB2 bind was done using the DBB User Build capability. Again the VTP JCL is submitted for batch execution and we can verify the bug on the batch output execution.

**On Section 4** -You execute a CICS NEWCOPY and use the HCAZ transaction again to verify the bug introduced by your COBOL modified program.

**On Section 5** -You fix the bug introduced and rebuild the program using IDz.

**On Section 6** -You run again the VTP JCL and verify that the bug is fixed. No more data mismatching from the original VTP recording.



**Congratulations!** You have completed the Exercise.