

IBM watsonx Code Assistant for Z: Execute Understand and Refactor phases on zVA ecosystem.

Collated and prepared by:

- Mageswaran A
- Ruta Thale

Mainframe Application Modernization Advisory Services, HCT

Reviewed By

- Ashish Sahoo
- Joydeep Banerjee

Table of Contents

| | | |
|----|--|----|
| 1. | Purpose of the document | 3 |
| 2. | How to login into ZVA..... | 3 |
| 3. | How to execute the Understand Phase using Application Discovery Tool on WCA4Z..... | 4 |
| 4. | How to execute the Refactor phase using IBM watsonx Code Assistant for Z Refactoring | 10 |
| | Assistant..... | 10 |

1. Purpose of the document

The purpose of this document is to showcase the capability of IBM Watsonx Code Assistant for Z to execute Understand, Refactor and Transform phases on zVA with a demo project. This document contains set of instructions, which if followed step by step will give enough knowledge and confidence to use IBM Watsonx Code Assistant for Z in any project. Changes in instructions sequence execution may be required when used in project depending on the environment setup.

As strategic enabler for incremental modernization, IBM Watsonx Code Assistant for Z Refactoring Assistant can help IBM Z Customers succeed in small modernization efforts and drive wider modernization in place.

IBM Watsonx Code Assistant for Z Refactoring Assistant can be used by:

- Architects
- Business Analysts
- Developers

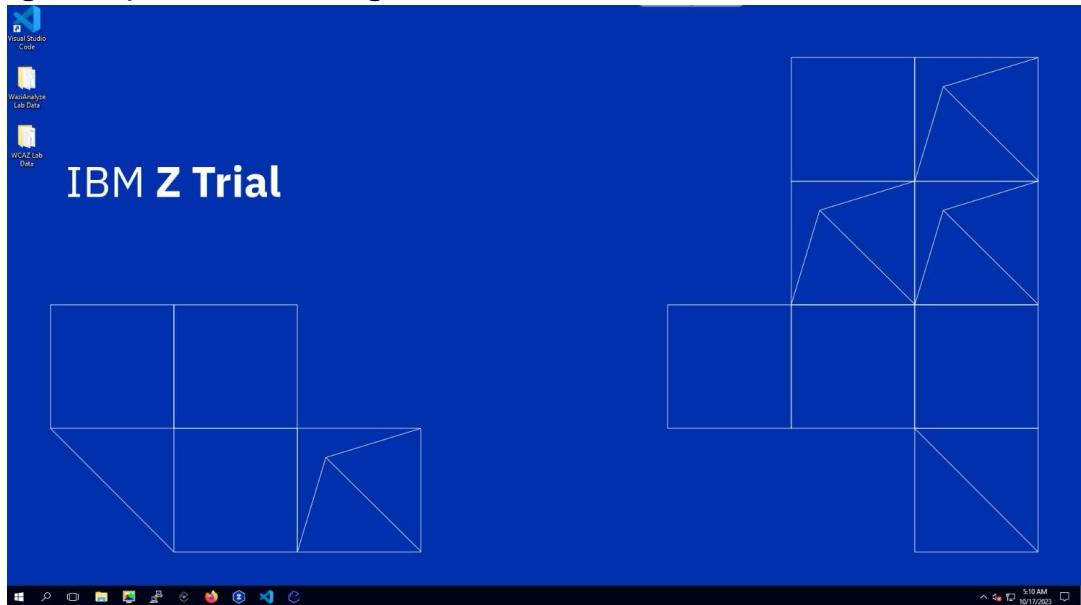
This document will help anyone to use the IBM Watsonx Code Assistant for Z, that helps developers identify the part of the application to refactor into modular and reusable services.

2. How to login into ZVA

1. Get your access for workshop link provided prior to the workshop.

The screenshot shows a login interface for the IBM Z Trial. At the top, a blue header bar displays the text "IBM Z Trial". Below this, the main content area has a white background. It features a heading "Welcome to IBM Z Trial." followed by a paragraph of text: "Thank you for choosing to try out ADDI-Watsonx Code Assists for Z. You can find your user credentials in the last email we sent you." Another paragraph states: "This demo will be active for 3 days from the date of the same email." A third paragraph provides contact information: "If you have any problems or you wish to extend the trial please contact ztrial@uk.ibm.com". Below the text, there are two input fields: one for "Username" and one for "Password", both with placeholder text "Username" and "Password" respectively. At the bottom of the form, there are two buttons: "Cancel" on the left and "Sign in" on the right, with the "Sign in" button being blue and highlighted.

2. Login with your credentials to get into ZVA.



3. How to execute the Understand Phase using Application Discovery Tool on WCA4Z

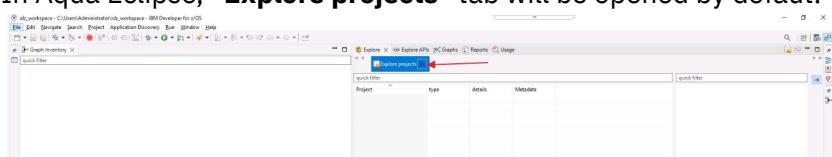
1. Close the IBM ztrial page opened on login

A screenshot of the "IBM Application Discovery and Delivery Intelligence" application. The title bar says "IBM Z Trial". The main interface features three cards: "SCENARIO | 30 MINS Discover candidate APIs in your code base", "SCENARIO | 30 MINS Discover program code that can be re-factored into a service", and "SCENARIO | 30 MINS Analyze source code before and after refactoring". Each card has a "Explore scenario" button. On the right, there's a "Related products" section with a link to learn about other products.

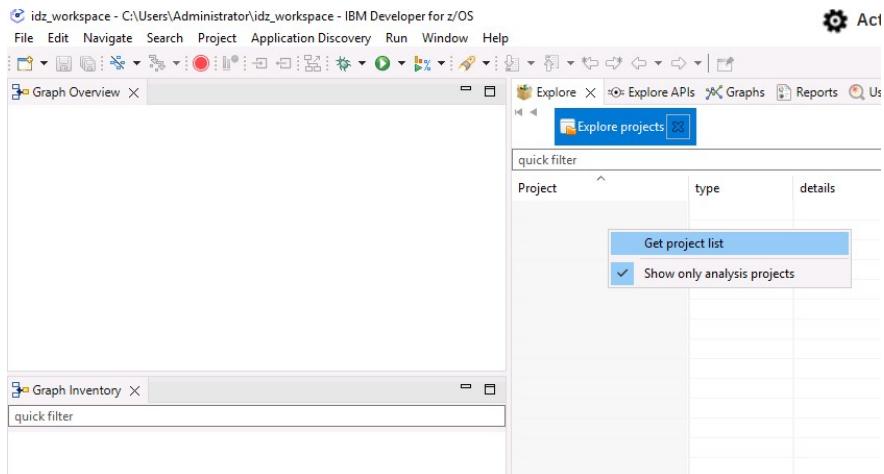
2. Open Aqua Eclipse on RDP browser from taskbar



3. In Aqua Eclipse, “Explore projects” tab will be opened by default.



4. Right click in the blank space below “Explore projects” and Select option “Get project list”.

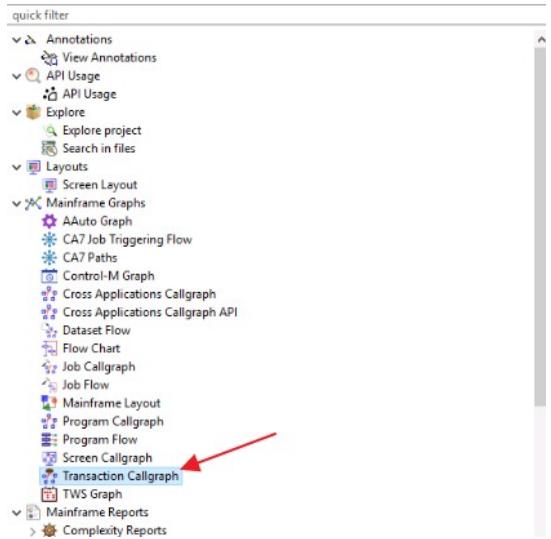


5. All projects will be listed. Select “GenApp”

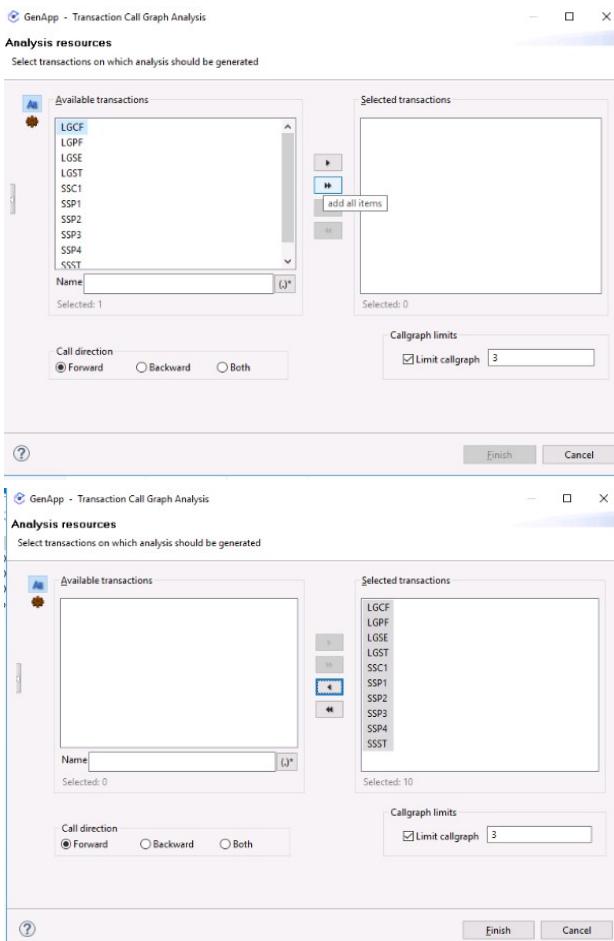
| Project | type | details | Metadata |
|--------------|------|-----------------|----------|
| GenApp | z/OS | MVS; (ADS, A... | |
| Hospital_510 | z/OS | MVS; (ADS, A... | |
| JKEBank_V100 | z/OS | MVS; (ADS, A... | |
| JKEBank_V101 | z/OS | MVS; (ADS, A... | |
| MortgageApp | z/OS | MVS; (ADS, A... | |
| TEST | z/OS | MVS; (ADS, A... | |

6. Select project ‘GenApp’ and new tab with all options will appear at the right side

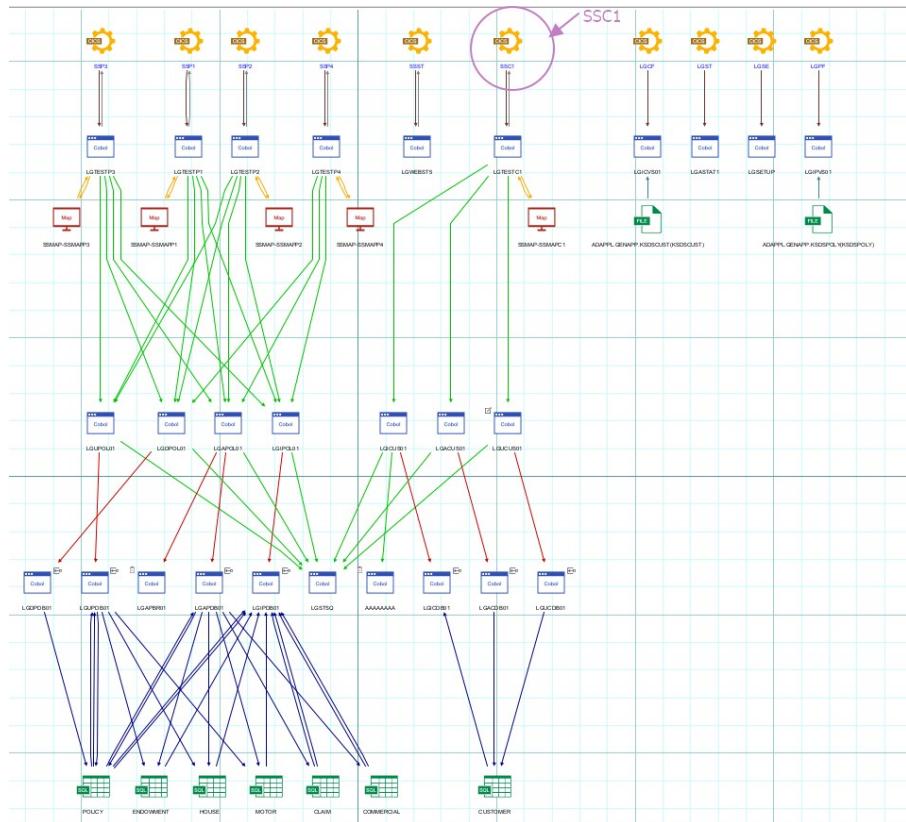
7. On the right side, **double click** to open option “Transaction Callgraph”



8. New pop-up window will open. Select all by using double triangle button and click on finish.



It will load the call graph for all transactions in the application.



The transaction SSC1 show in above diagram is related to Customer related functions. Transaction SSC1 is used to Create new Customer, Display Customer and Update Customer. SSC1 will be used in all next phases.

9. Click on Explorer tab to see following project list and options at the right side.

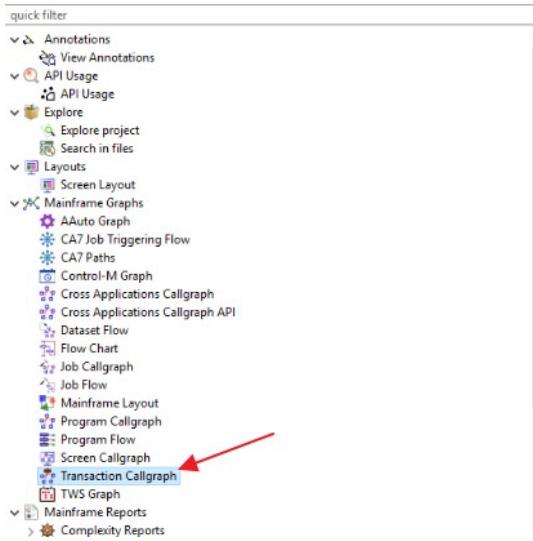
The screenshot shows the 'Explore' interface with the 'Explore' tab selected. On the left, there is a project list table:

| Project | Type | Details | Metadata |
|--------------|------|-----------------|----------|
| GenApp | z/OS | MVS; (ADS, A... | |
| Hospital_510 | z/OS | MVS; (ADS, A... | |
| JKBank_V100 | z/OS | MVS; (ADS, A... | |
| JKBank_V101 | z/OS | MVS; (ADS, A... | |
| MortgageApp | z/OS | MVS; (ADS, A... | |
| TEST | z/OS | MVS; (ADS, A... | |

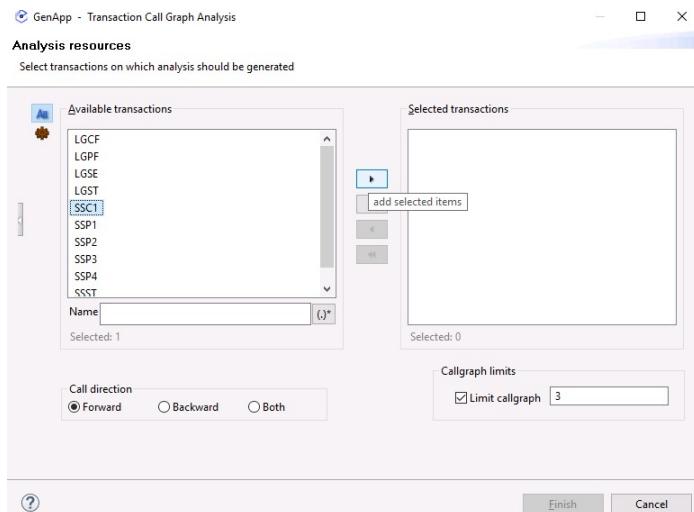
On the right, there is a sidebar with various navigation options:

- Annotations
- API Usage
- Explore
- Explore project
- Search in files
- Layouts
- Mainframe Graphs
- AAuto Graph
- CA7 Job Triggering Flow
- COBOL
- Control-M Graph
- Cross Applications Callgraph
- Cross Applications Callgraph API
- Dataset Flow
- Flow Chart
- Job Callgraph
- Job Flow
- Mainframe Layout
- Program Callgraph
- Program Flow
- Screen Callgraph
- Transaction Callgraph

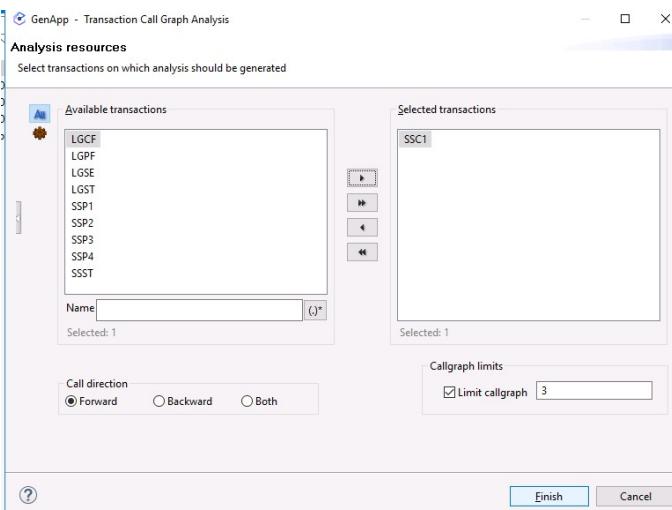
10. On the right side, **double click** to open option “Transaction Callgraph”



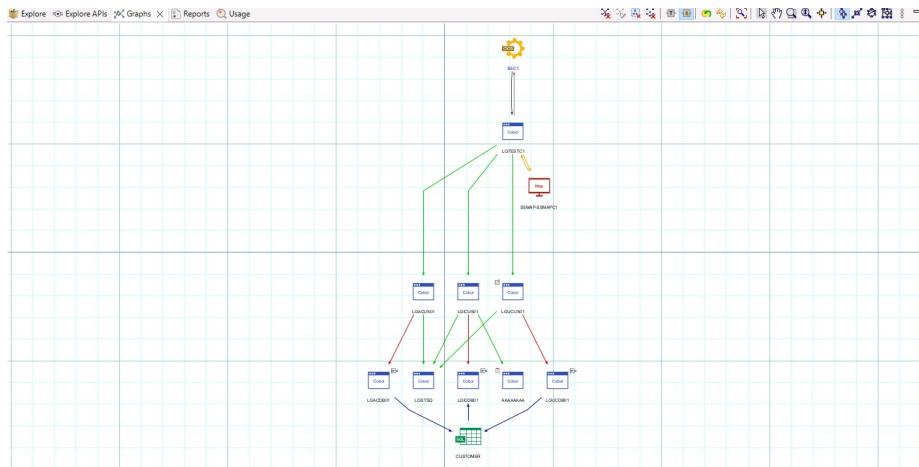
11. On the new Pop-up, select SSC1 transaction and click on single triangle to add the transaction.



12. Click Finish to load the transaction SSC1.

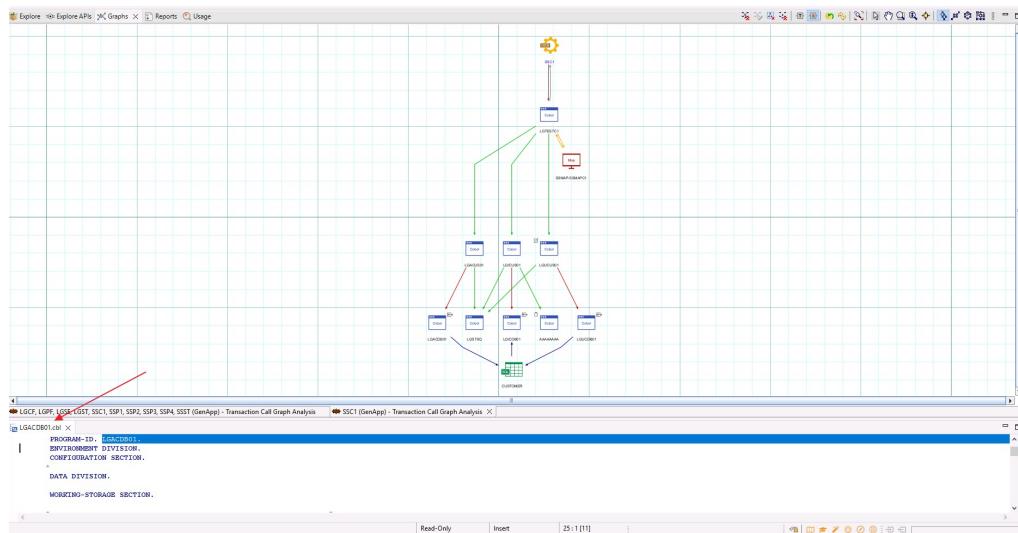


13. Transaction graph will be displayed.



- a. In the transaction callgraph of transaction SSC1, there are different components.
- i. CICS transaction SSC1
 - ii. Main Transaction program LGTESTC1. this program calls map SSMAPC1
 - iii. Depending on the function selected by user, it can perform different operations.
 1. Add/Insert Customer (LGACUS01)
 2. Inquire Customer (LGICUS01)
 3. Update Customer (LGUCUS01)
 - iv. These programs call the data access programs to interact with table “Customer”.
 1. Add/Insert Customer (LGACDB01)
 2. Inquire Customer (LGICDB01)
 3. Update Customer (LGUCDB01)

14. Double click on the White and Blue box named COBOL to open the respective code. e.g., LGACDB01. Code will open at the bottom tab.



15. Drag the tab with code LGACDB01 to the right side to see parallel view of code and graph. Close all open Tabs other than tabs with graph and code .

LGACDB01 code

In the code LGACDB01, we can see the main function to insert the customer record into the Customer table.

```

SSC1 (GenApp) - Transaction Call Graph Analysis
SSC1 (GenApp) - Transaction Call Graph Analysis

INSERT-CUSTOMER.
-----+
+ Insert row into Customer table based on customer number
-----+
MOVE *'INSERT CUSTOMER' TO EM-SQLSQ
-----+
IF LGAC-NCH = 'ON'
EXEC SQL
  INSERT INTO CUSTOMER
    (CUSTOMERNUMBER,
     FIRSTNAME,
     LASTNAME,
     DATERFIRST,
     HOUSENUMBER,
     POSTCODE,
     PHONEHOME,
     PHONEWORK,
     EMAILADDRESS )
  VALUES (:LGZ-CUSTOMERNUM-INT,
          :CA-FIRST-NAME,
          :CA-LAST-NAME,
          :CA-DATEFIRST,
          :CA-HOUSENUMBER,
          :CA-POSTCODE,
          :CA-PHONEHOME,
          :CA-PHONEWORK,
          :CA-EMAIL-ADDRESS )
-----+
END-EXEC
IF SQLCODE NOT EQUAL 0
  MOVE *90 TO CA-RETURN-CODE
  PREPARE WHILE-ERROR-MESSAGE
  EXEC CICS RETURN END-EXEC
-----+
ELSE-IF
EXEC SQL
  INSERT INTO CUSTOMER
    (CUSTOMERNUMBER,
     FIRSTNAME,
     LASTNAME,
     DATERFIRST,
     HOUSENUMBER,
     POSTCODE,
     PHONEHOME,
     EMAILADDRESS )
  VALUES (:LGZ-CUSTOMERNUM-INT,
          :CA-FIRST-NAME,
          :CA-LAST-NAME,
          :CA-DATEFIRST,
          :CA-HOUSENUMBER,
          :CA-POSTCODE,
          :CA-PHONEHOME,
          :CA-EMAIL-ADDRESS )
-----+
END-IF
-----+
      
```

In this **Understand** phase,

1. we checked callgraph of all the transactions in the application and selected SSC1 transaction to understand the flow and related components.
2. We checked the data access program LGACDB01 to insert customer in data table **Customer**.

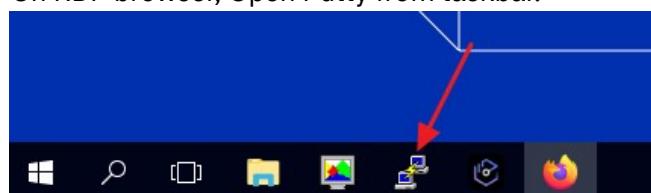
We will use this knowledge in next steps to Refactor and Transform.

4. How to execute the Refactor phase using IBM watsonx Code Assistant for Z Refactoring Assistant

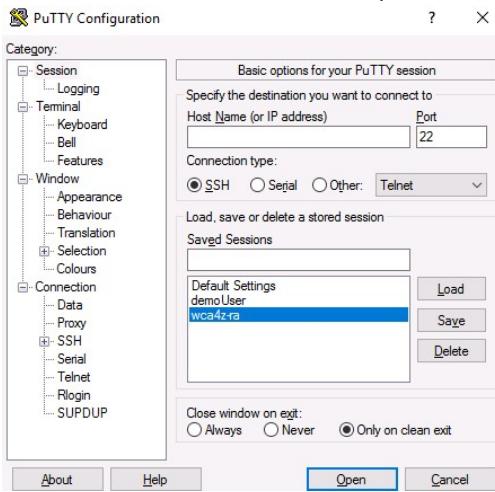
The initial installation and environment level setup will be done by the system programmer. Below initial setup steps are for the user who will be using the IBM watsonx Code Assistant for Z Refactoring Assistant in zVA environment.

Procedures to access refactoring assistant on zVA

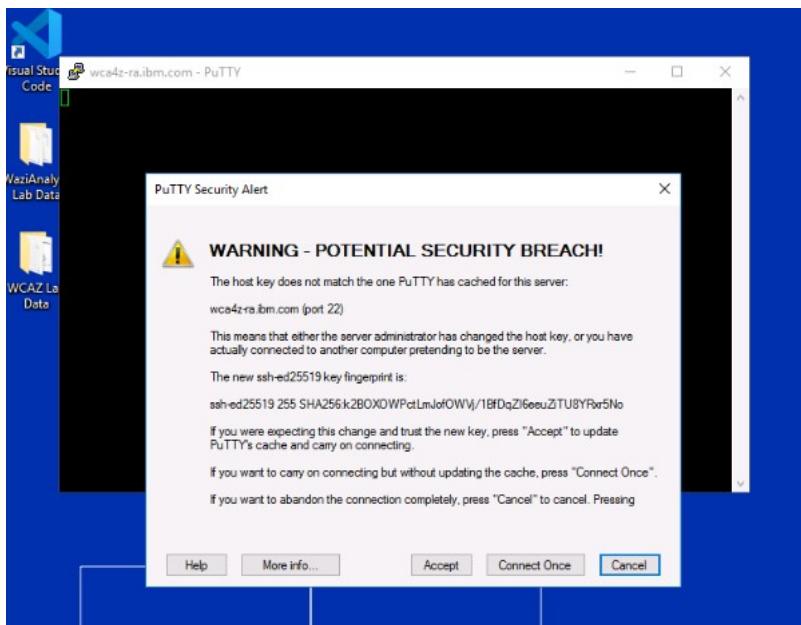
1. On RDP browser, Open Putty from taskbar.



2. select wca4z-ra and click on open.



3. After clicking on Open in previous step following warning will pop-up. Select "Accept". This will be displayed only first time.



4. Auto scripts will run on Putty to start the Refactoring Assistant.

Wait till message “IBM Watson code assistant for Z Refactoring Assistant started” is displayed on putty. After this message close putty using close button on right top.

```

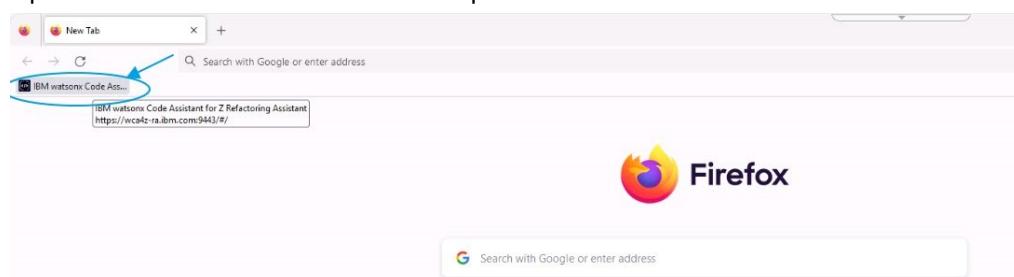
demo@t-10393-k-x86: ~
What is the OrientDB password? This password will be required every time IBM watsonx Code Assistant for Z Refactoring Assistant is started.
=====
=====
Starting OrientDB...
/home/demo/.config/cni/net.d/sd_network.conflist
ed027364a5e316cdf4e8f429577cad4dd9424d1f843bf990fe5563d09b8f58b9
Bundling orientdb.crt and orientdb.key into pkcs12 keystore
Restarting OrientDB with SSL configuration
ed027364a5e316cdf4e8f429577cad4dd9424d1f843bf990fe5563d09b8f58b9
6504f44973d867e21ce139a5e2da9eb06923bf32e54f4c776935b4e64198e851
Starting IBM watsonx Code Assistant for Z Refactoring Assistant...
Bundling ad-core-server.crt and ad-core-server.key into pkcs12 keystore
Certificate was added to keystore
Bundling service-designer.crt and service-designer.key into pkcs12 keystore
Certificate was added to keystore
Certificate was added to keystore
Certificate was added to keystore
f25c66c54805ad934cfde6c431871c7c28e014588055cfffa646e4b70ad84853
38f0c6d737e4ba310dd5843079elac8524d1abfe4d64e8299959c7560068513a
c598ef6a0970d7733ad57572c5466c2574bb037845a2292ed7fbe6b2f8b3c657
IBM watsonx Code Assistant for Z Refactoring Assistant started
demo@t-10393-k-x86:~$ 

```

5. Open Firefox from Taskbar.



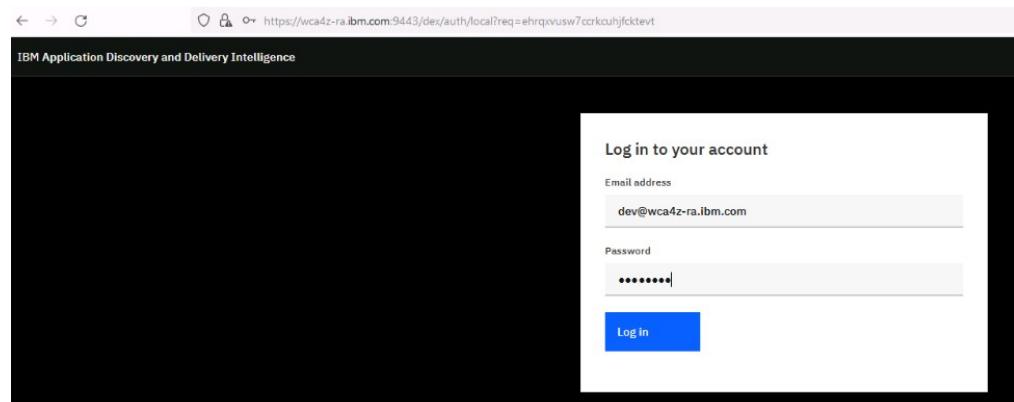
6. Open the link from Favourite Bar to open the IBM watson Code Assistant for z.



7. Login with following credentials

ID: dev@wca4z-ra.ibm.com

Password: password



8. After login, you can see following tabs.
 - a. My Workspace
 - b. Workspace shared with me.

"My Workspace" tab will be opened by default.

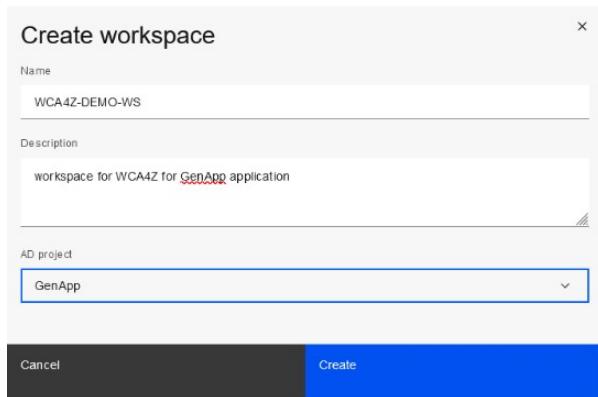
After you build a project in Understand phase, you can create a workspace to work on that project with IBM watsonx Code Assistant for Z Refactoring Assistant.

9. Under My workspace tab, click on the '**Create workspace**' dialogue box.

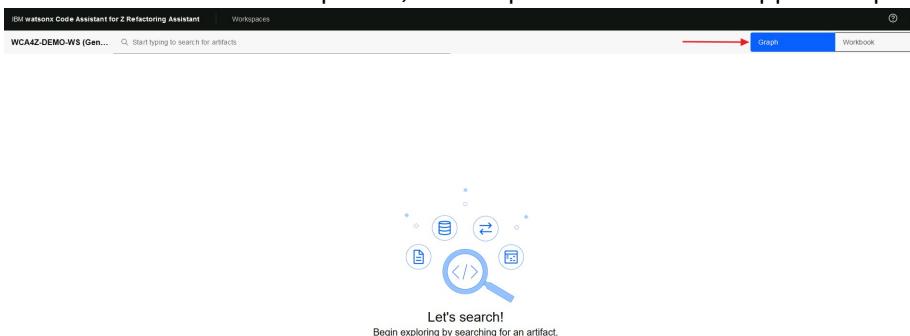
10. In the "Create workspace" dialog

- a. Type a name for the workspace that you want to create, for example, GenApp.
 - i. Name is given as 'WCA4Z-DEM-WS' here.
- b. Provide description, e.g., workspace for WCA4Z for GenApp application.
- c. From the 'AD Project' dropdown list, select the project that you built with IBM AD Build Client
 - i. 'GenApp' selected as AD Project

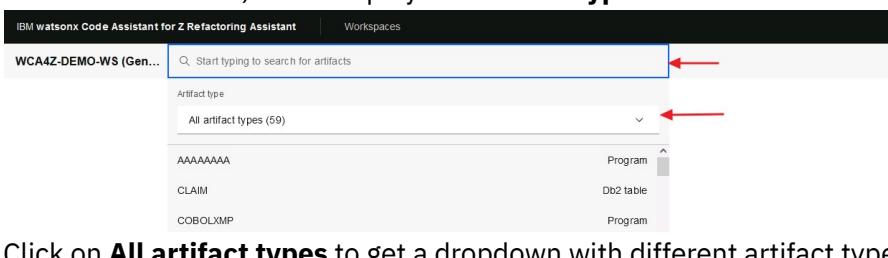
11. Click '**Create**' to create the workspace.



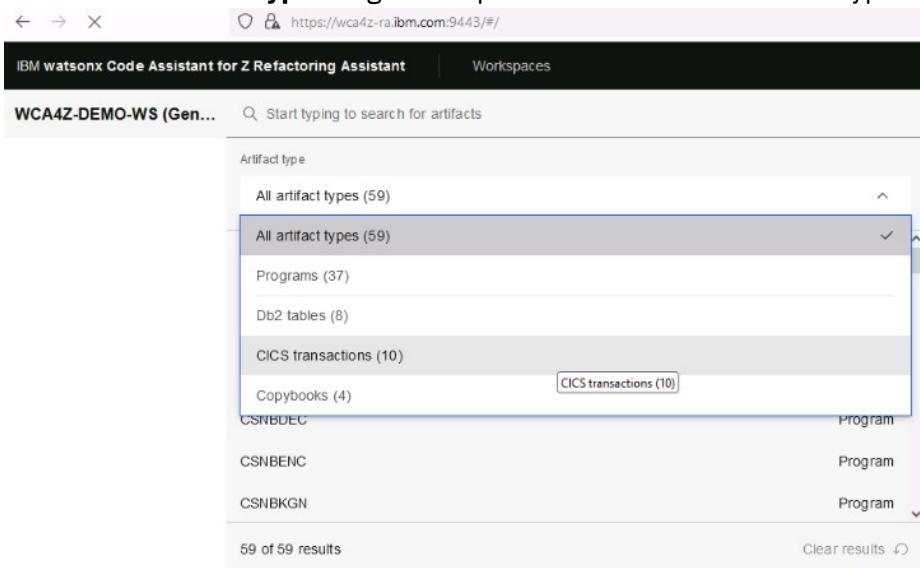
12. When the creation is completed, the Graph view of the GenApp workspace is displayed.



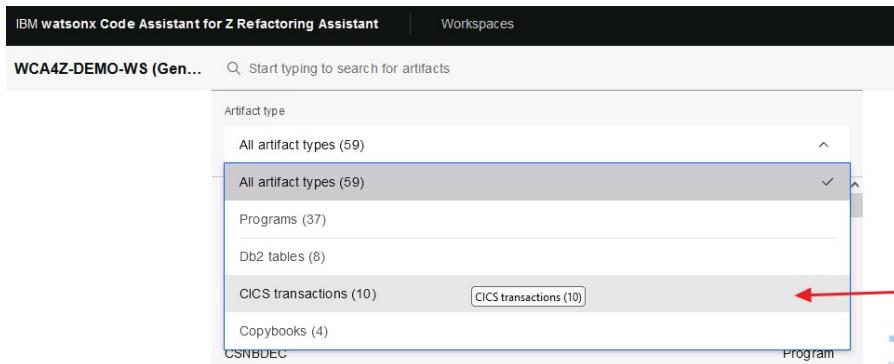
13. Click on Search bar, it will display '**all artifact types**'.



14. Click on **All artifact types** to get a dropdown with different artifact types.



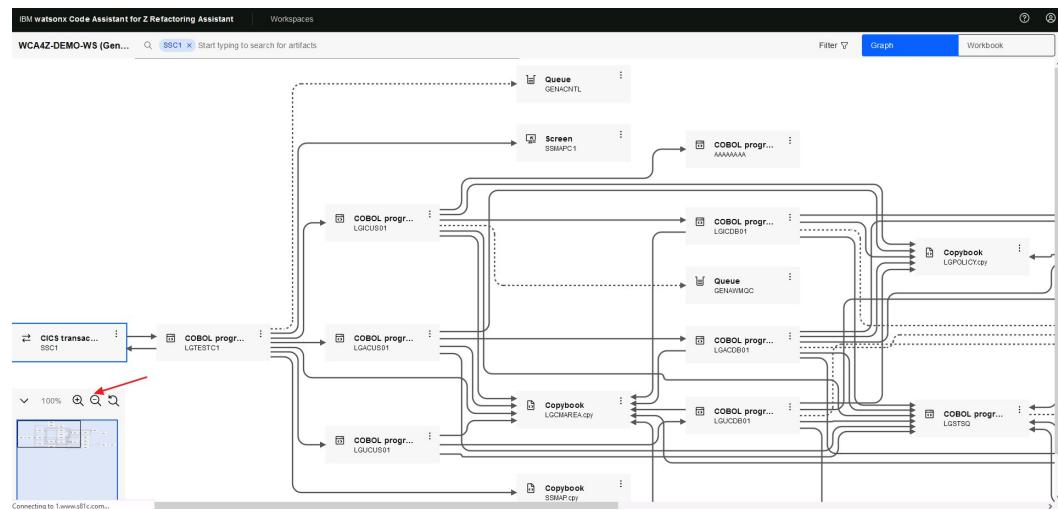
15. Select CICS transactions.



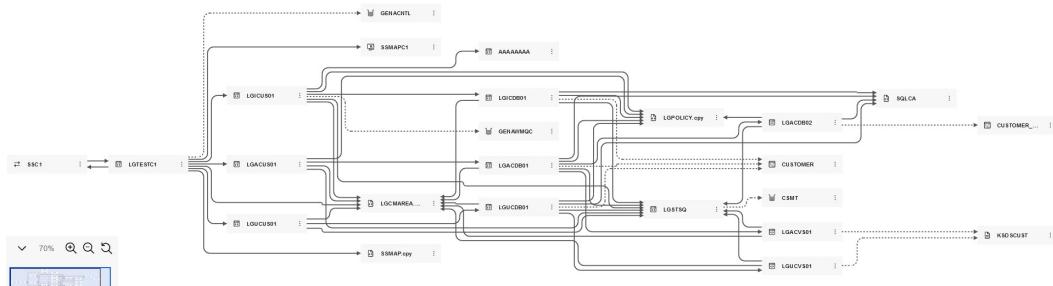
16. From the dropdown list of transactions, click on SSC1 and press enter.

| Transaction | Type |
|-------------|------------------|
| LGCF | CICS transaction |
| LGPF | CICS transaction |
| LGSE | CICS transaction |
| LGST | CICS transaction |
| SSC1 | CICS transaction |
| SSP1 | CICS transaction |
| SSP2 | CICS transaction |

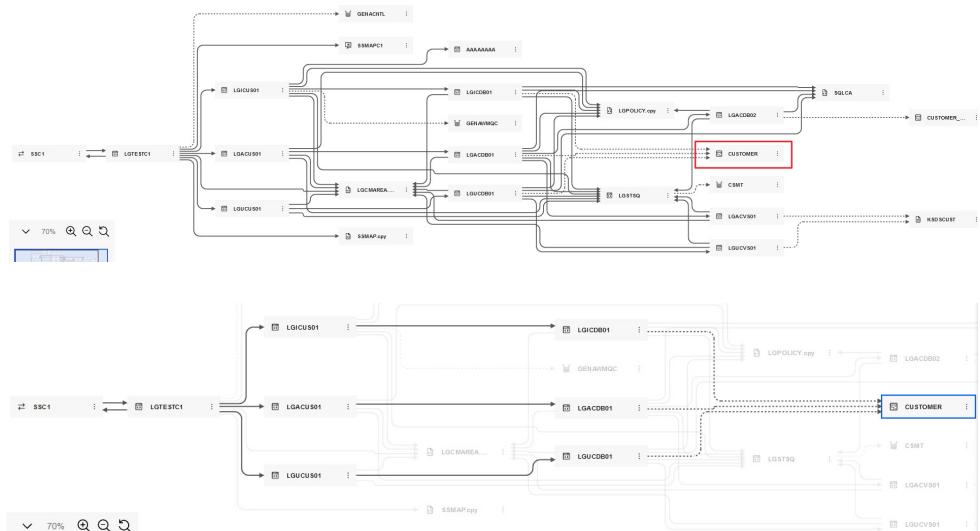
17. Graph for SSC1 transaction will open. Zoom out using zoom out icon to see complete callgraph.



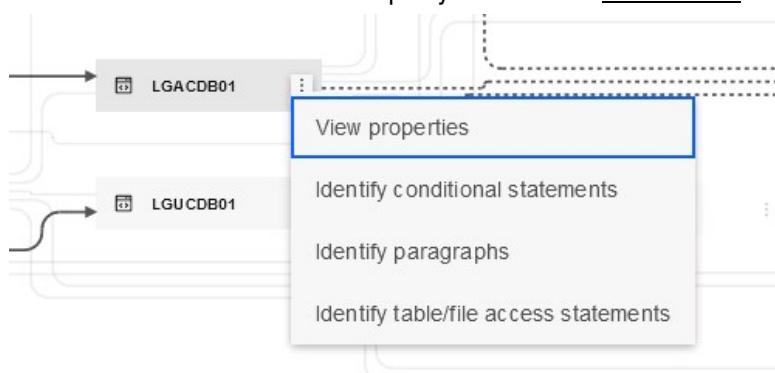
18. Graph after zooming out will be displayed.



19. Click on the Customer Table to get the flow of program dependencies for Customer table.



20. In the Understand phase, we checked the callgraph for SSC1 transactions and we checked the insert Customer query in the code [LGACDB01](#).



Click on the 3 dots/ellipses besides the [LGACDB01](#) to get below options.

- View properties
- Identify conditional statements.
- Identify paragraphs.
- Identify tables/file access statements.

21. Click on **Identify conditional statements** to display all conditional statements in the code in the order of importance/complexity.

```

Key conditions
Select a key condition type
Conditional statements

15 conditional statements ordered by
importance

133
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187

* initialize DB2 host variables
INITIALIZE DB2-OUT-INTGERS.

*-
* Process incoming commarea
*-
* If NO commarea received issue an ABEND
IF EIBCALEN IS EQUAL TO ZERO
MOVE 'NO COMMAREA RECEIVED' TO EM-VARIABLE
PERFORM WRITE-ERROR-MESSAGE
EXEC CICS ABEND ABCODE('LGCA') NODUMP END-EXEC
END-IF

* initialize commarea return code to zero
MOVE '00' TO CA-RETURN-CODE
MOVE EIBCALEN TO WS-CALEN.
SET WS-ADDR-DFHCOMMAREA TO ADDRESS OF DFHCOMMAREA.

* check commarea length
ADD WS-CA-HEADER-LEN TO WS-REQUIRED-CA-LEN
ADD WS-CUSTOMER-LEN TO WS-REQUIRED-CA-LEN

* if less set error return code and return to caller
IF EIBCALEN IS LESS THAN WS-REQUIRED-CA-LEN
MOVE '98' TO CA-RETURN-CODE
EXEC CICS RETURN END-EXEC
END-IF

* Call routine to Insert row in Customer table
PERFORM Obtain-CUSTOMER-Number.
PERFORM INSERT-CUSTOMER.

EXEC CICS LINK Program(LGACVS01)

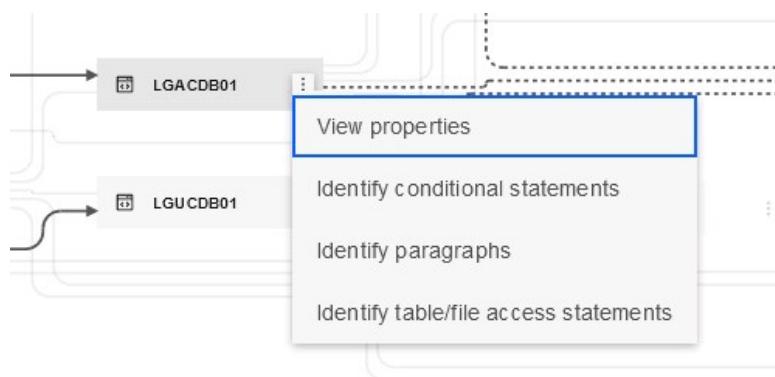
```

22. Click on “Graph” on the right top corner to return to Graph view.

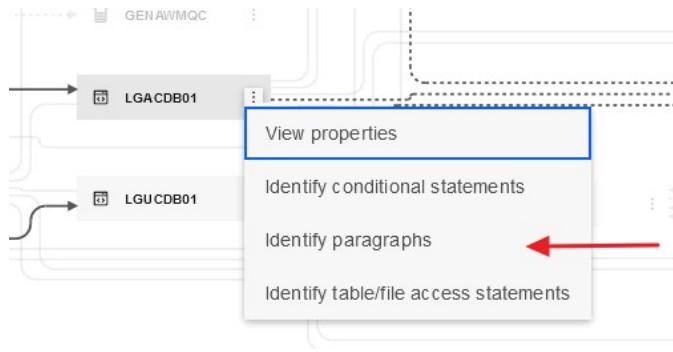


23. Click on the 3 dots/ellipses besides the LGACDB01 to get below options

- View properties
- Identify conditional statements.
- Identify paragraphs.
- Identify tables/file access statements.



24. Click on “Identify paragraphs” option.



25. It will switch from graph view to Workbook view with

In Workbook view you can see 3 sections

- Workbook details on extreme left section
- All paragraphs in the code in the order of importance/complexity in the middle section
- Code is displayed on extreme right side.

26. As we are checking Insert Customer function, Click on **INSERT-CUSTOMER** paragraph under Key conditions to see the paragraph in the code.

27. Right Click on the paragraph name **INSERT-CUSTOMER** appearing in the code you will get option “Slice on paragraph to new workbook”. Click on this option.

Key conditions

Select a key condition type

Paragraphs

5 paragraphs ordered by importance

MAINLINE SECTION. (line 141)

INSERT-CUSTOMER. (line 232)

WRITE-ERROR-MESSAGE. (line 312)

Obtain-CUSTOMER-Number. (line 216)

MAINLINE-EXIT. (line 211)

```

223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238

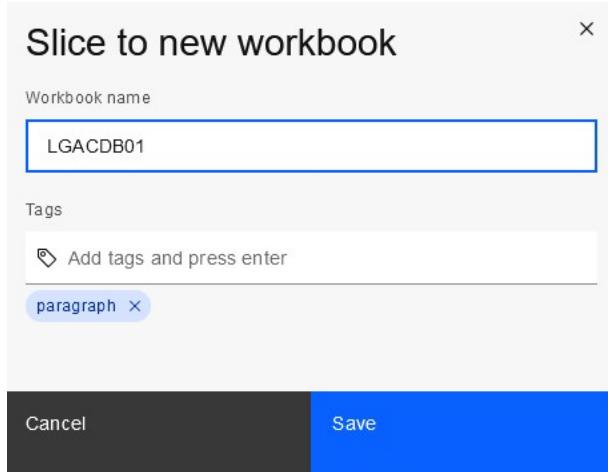
```

IT WS->RESP-NUL = UPRRESP(NORMAL)
MOVE 'NO' TO LGAC-NCS
Initialize DB2-CUSTOMERNUM-INT
ELSE
Move LastCustNum To DB2-CUSTOMERNUM-INT
End-If.

INSERT-CUSTOMER.

*-----
* Inse-----
*-----
MOVE ' INSERT CUSTOMER' TO EM-SQLREQ
*-----
IF LGAC-NCS = 'ON'

28. New pop-up will appear, give name as LGACDB01, and click on Save.



29. Code will be sliced into new workbook which will be displayed at the left side section. And selected code will be marked with dotted line on right side section.

Workbooks /

Workbook details

Name: LGACDB01 (New) Status: New

Service code blocks for LGACDB01

2 code blocks ●

Code block from LGACDB01 (lines 232-304) paragraph

INSERT-CUSTOMER.

* Insert row into Customer table based on customer number

Code block from LGACDB01 (lines 312-345) paragraph

WRITE-ERROR-MESSAGE.
* Save SQLCODE in message
MOVE SQLCODE TO EM-SQLRC
* Obtain and format current time and date

LGACDB01 x

```

223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263

```

IT WS->RESP-NUL = UPRRESP(NORMAL)
MOVE 'NO' TO LGAC-NCS
Initialize DB2-CUSTOMERNUM-INT
ELSE
Move LastCustNum To DB2-CUSTOMERNUM-INT
End-If.

INSERT-CUSTOMER.

*-----
* Insert row into Customer table based on customer number *-----

MOVE ' INSERT CUSTOMER' TO EM-SQLREQ
*-----
IF LGAC-NCS = 'ON'
EXEC SQL
INSERT INTO CUSTOMER
(CUSTOMERNUMBER,
FIRSTNAME,
LASTNAME,
DATEOFBIRTH,
HOUSENAME,
HOUSENUMBER,
POSTCODE,
PHONEMOBILE,
PHONEHOME,
EMAILADDRESS)
VALUES (:DB2-CUSTOMERNUM-INT,
:CA-FIRST-NAME,
:CA-LAST-NAME,
:CA-DOB,
:CA-HOUSE-NAME,
:CA-HOUSE-NUM,
:CA-POSTCODE,
:CA-PHONE-MOBILE,
:CA-PHONE-HOME,
:CA-EMAIL-ADDRESS)
END-EXEC
IF SQLCODE NOT EQUAL 0
MOVE '99' TO CA-RETURN-CODE

30. In the Workbook details section, click on the 3 dots/ellipses besides the **new** and select option “Export” to export the sliced code.

Workbook details

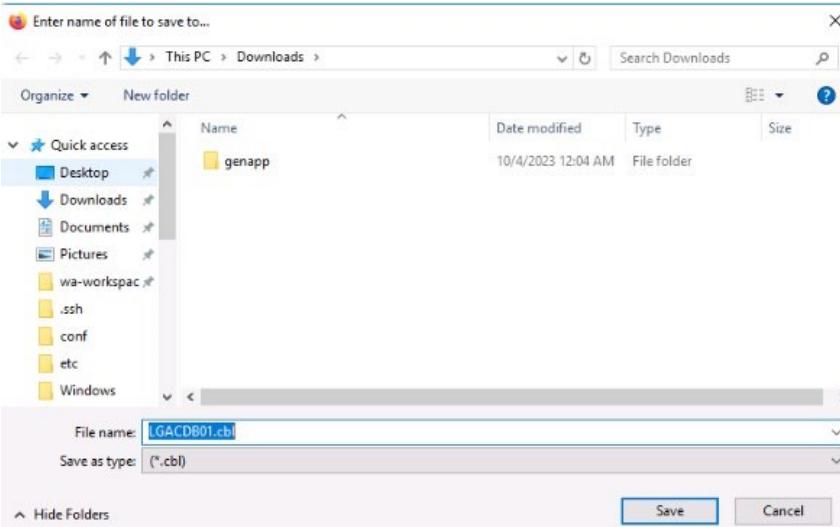
Name: LGACDB01 (New) Status: New

Service code blocks for LGACDB01

2 code blocks Add code block Import Export Rename Delete

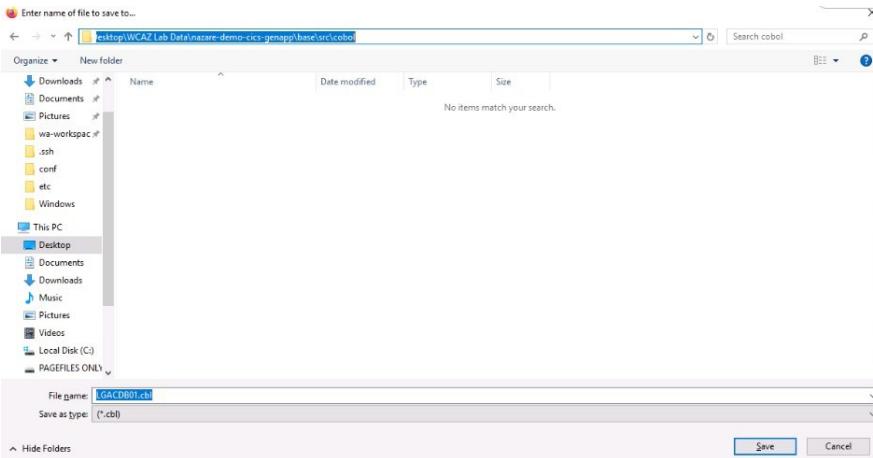


31. Windows folder pop-up will appear with default location to save the exported sliced code.



32. Browse to the following path and click on “Save” button to save the exported code to correct location. This is one time step required for Export.

folder path: C:\Users\Administrator\Desktop\WCAZ Lab Data\nazare-demo-cicsgenapp\base\src\cobol



In this **Refactor** phase,

1. We used information from Understand phase to select SSC1 transaction.
2. We used IBM watsonx Code Assistant for Z Refactoring Assistant for refactoring the insert customer functionality by slicing code LGACDB01 into workbook.
3. We exported this workbook to be used in next Transform phase.