

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

Collated and prepared by:

- Vivek Kakkar

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 watson Code Assistant for Z (WCA4Z)	4
4. How to execute the Refactor phase using IBM watsonx Code Assistant for Z Refactoring Assistant	11
5. Explanation feature of WCA4Z using VS Code	Error! Bookmark not defined.
6. How to execute the Transform Phase using VS Code on WCA4Z	Error! Bookmark not defined.
7. How to execute the Validate Phase using VS Code on WCA4Z	Error! Bookmark not defined.
8. Queries & Issues	Error! Bookmark not defined.

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 Z Virtual Access (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 clients 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

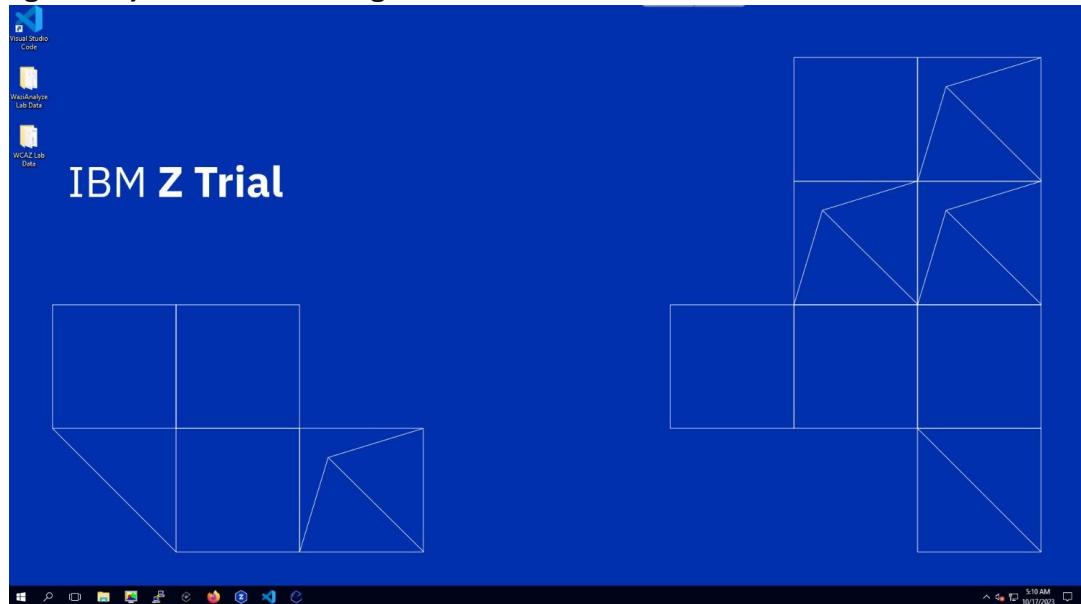
This document will help anyone to use the IBM watsonx Code Assistant for Z, which helps developers identify the part of the application to refactor into modular and reusable services.

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." Further down, 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 fields, there are two input fields: "Username" and "Password". The "Password" field is highlighted with a blue border. At the bottom of the form, there are two buttons: "Cancel" on the left and "Sign in" on the right, both in white text on a dark background.

2. Login with your credentials to get into ZVA.



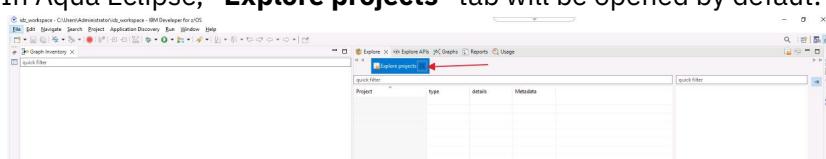
3. How to execute the Understand Phase using Application Discovery Tool on watson Code Assistant for Z (WCA4Z)

1. Close the IBM ztrial page opened on login

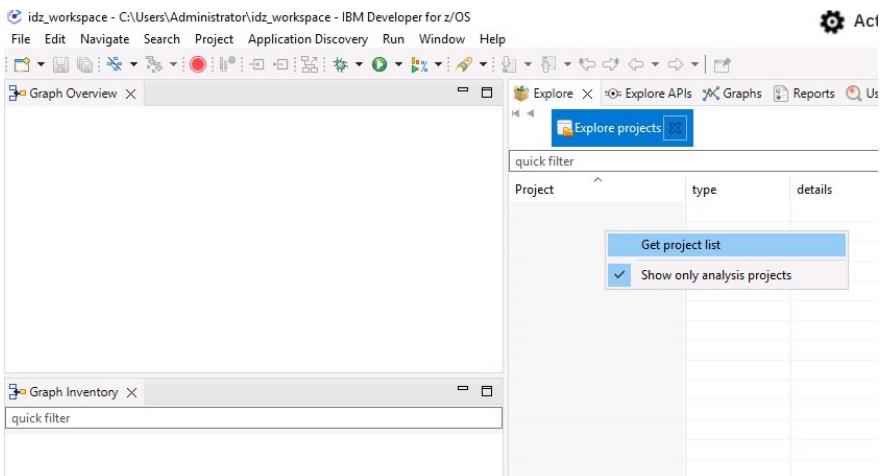
2. Open IBM Explorer on RDP browser from taskbar



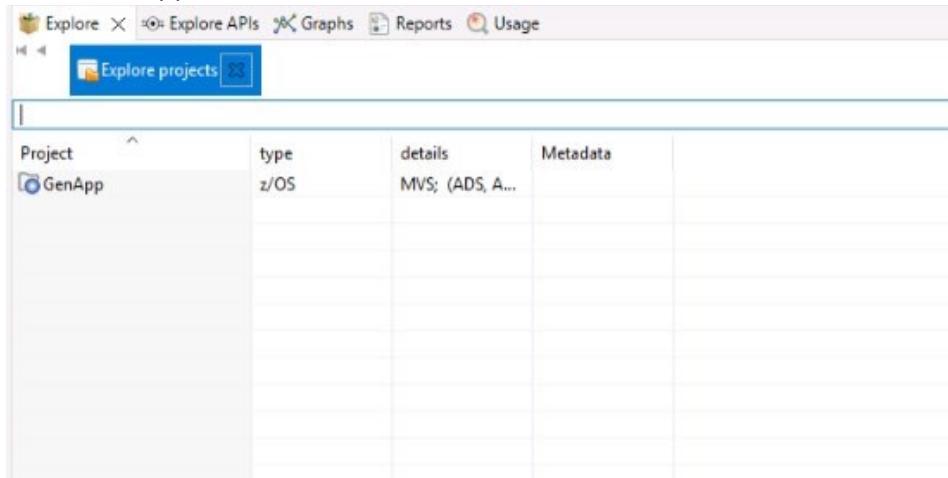
3. In Aqua Eclipse, “Explore projects” tab will be opened by default.



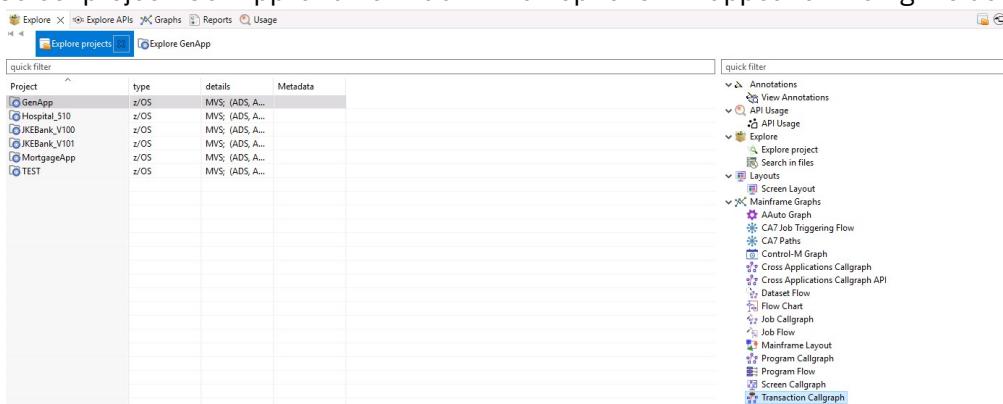
4. Right click in the blank space below “Explore projects” and Select option “Get project list”.

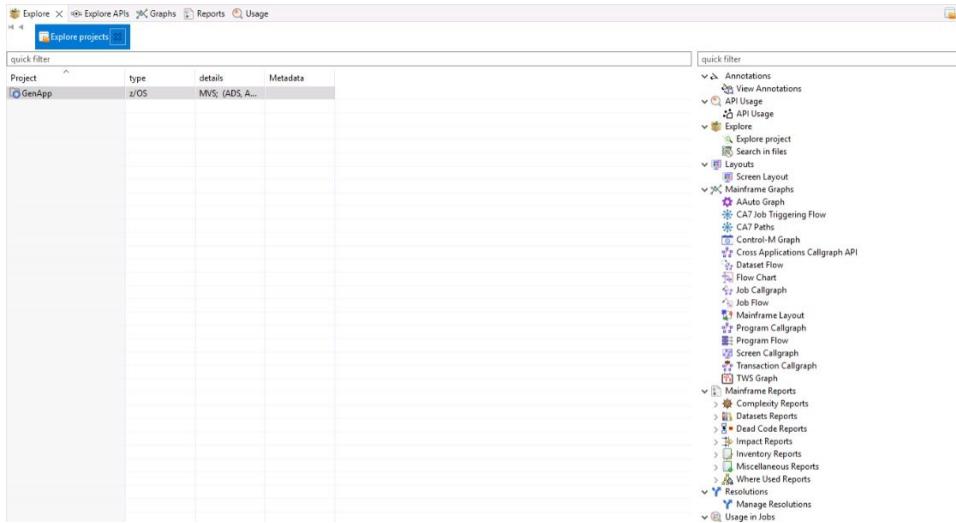


5. Select “GenApp”.

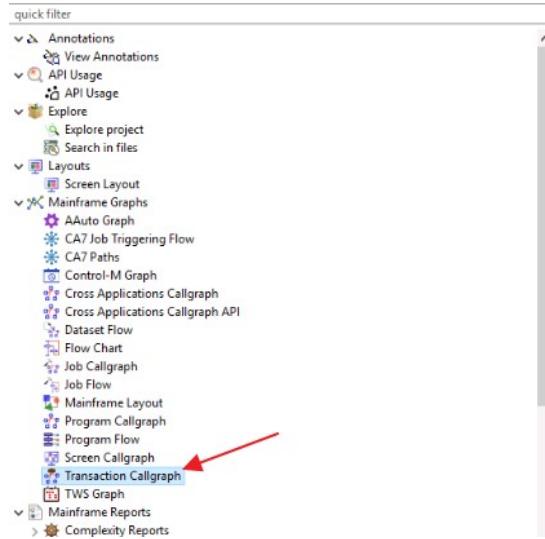


6. Select project ‘GenApp’ and new tab with all options will appear at the right side

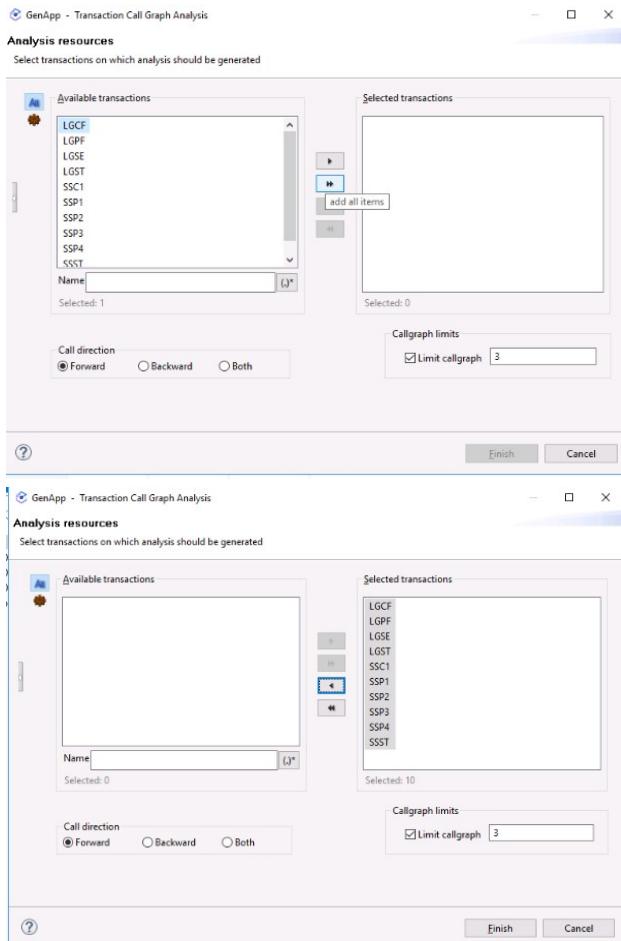




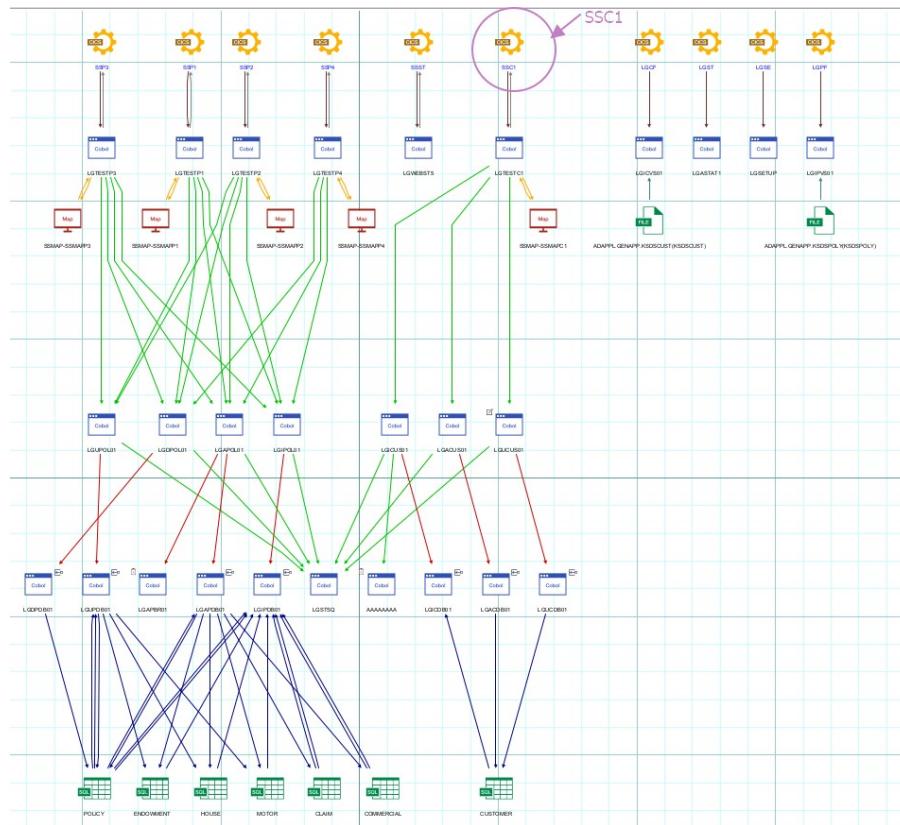
7. On the right side, **double click** to open option “Transaction Callgraph”



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



The call graph will be loaded for all transactions in the application. It may take few minutes for the graph to build.



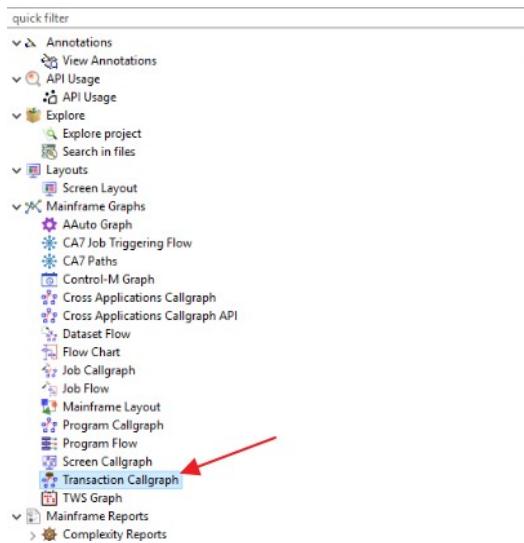
The transaction SSC1 shown in the above diagram is related to Customer related functions. Transaction SSC1 is used to Create new Customer, Display Customer and Update Customer. SSC1 will be used in all next phases.

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

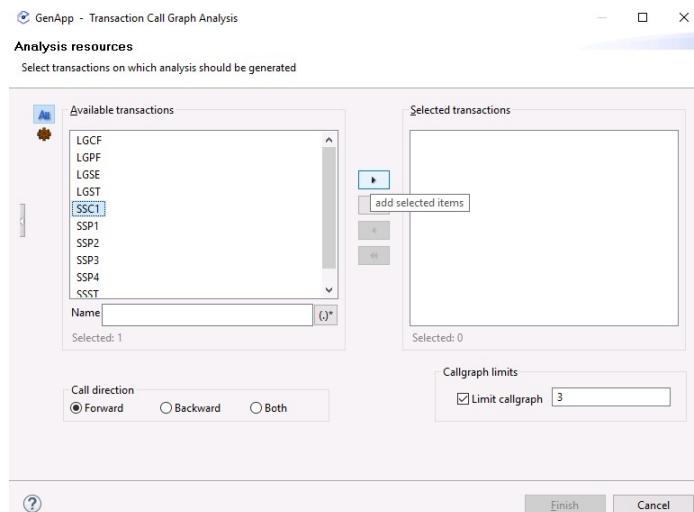
The screenshot shows the SAP ALE Explorer interface. The top navigation bar includes 'Explore', 'Explore APIs', 'Graphs', 'Reports', and 'Usage'. Below the navigation is a 'quick filter' search bar. The main area displays a table with columns 'Project', 'type', 'details', and 'Metadata'. One row is visible, representing a project named 'GenApp' of type 'z/OS' with details 'MVS; (ADS, A...'. To the right of the table is a 'quick filter' sidebar containing a tree view of graph types:

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

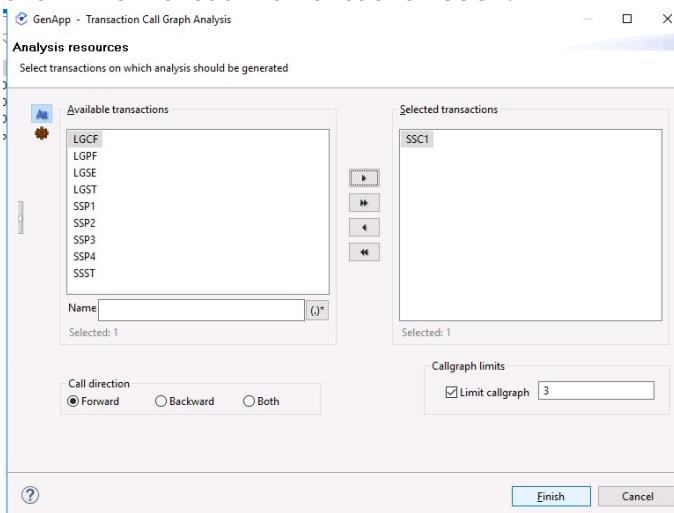
10. On the right side, **double click** to open option “Transaction Callgraph”



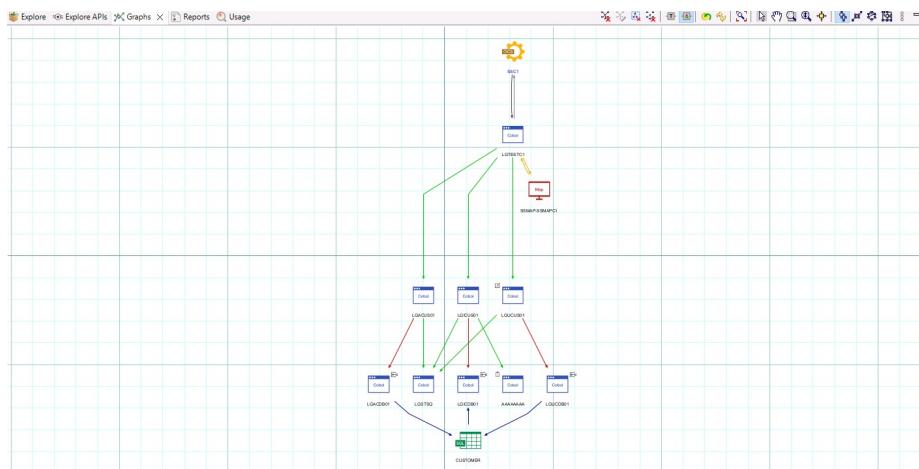
11. On the new Pop-up, select SSC1 transaction and click on single triangle to add the transaction.



12. Click Finish to load the transaction SSC1.

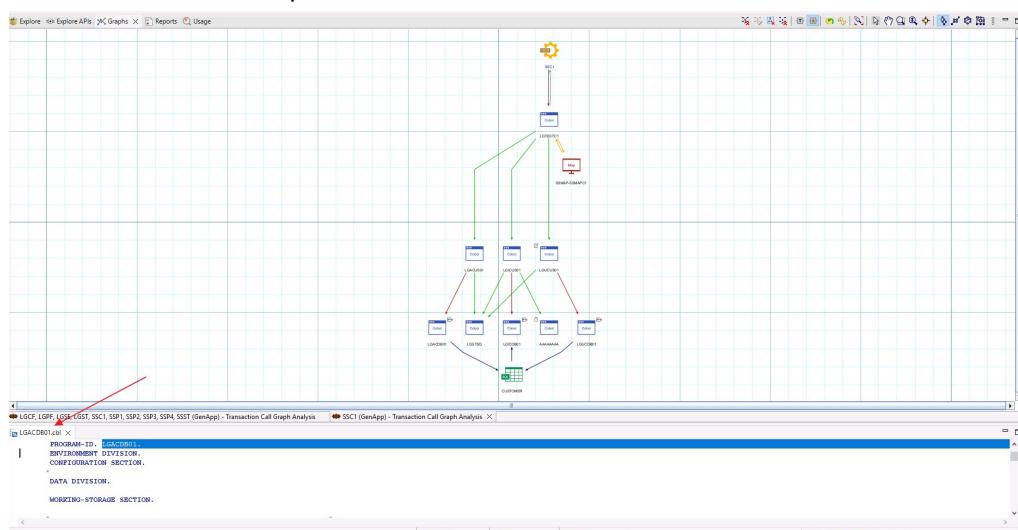


13. Transaction graph will be displayed.



- 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.

The screenshot shows the IBM Developer for z/OS interface. On the left, a transaction call graph displays various transactions like LGCF, LOPF, LSEI, LOST, SSC1, SSP1, SSP2, SSP3, SSP4, SSP5, and SSS1. A specific transaction, LGTEST1, is highlighted. On the right, the source code for LGACDB01 is displayed in a code editor window. The code is a COBOL program with SQL statements for inserting data into the CUSTOMER table. It includes logic to handle different SQL codes and return codes.

```

      * Insert row into Customer table based on customer number
      MOVE *'INSERT CUSTOMER' TO EM-SQRLQ
      IF LGAC-NCS = 'ON'
      EXEC SQL
      INSERT INTO CUSTOMER
      ( CUSTOMERNUM,
      FIRSTNAME,
      LASTNAME,
      DATESOFBIRTH,
      HOUSENUMBER,
      HOUSENAME,
      POSTCODE,
      PHONENUMBER,
      PHONENUMSILE,
      ADDRESS,
      EMAILADDRESS )
      VALUES ( IBS-CUSTOMERNUM-INT,
      :CA-FIRST-NAME,
      :CA-LAST-NAME,
      :CA-DOB,
      :CA-HOUSE-NUM,
      :CA-HOUSE-NAME,
      :CA-POSTCODE,
      :CA-PHONE-NUM,
      :CA-PHONE-SILE,
      :CA-EMAIL-ADDRESS,
      :CA-EMAIL-ADDRESS )

      END-EXEC
      IF SQLCODE NOT EQUAL 0
      MOVE *'P' TO CA-RETURN-CODE
      PERFORM UNTIL CA-RETURN-CODE
      EXEC CICS RETURN END-EXEC
      END-IF
      ELSE
      EXEC SQL
      INSERT INTO CUSTOMER
      ( CUSTOMERNUM,
      FIRSTNAME,
      LASTNAME,
      DATESOFBIRTH,
      HOUSENUMBER,
      HOUSENAME,
      POSTCODE,
      PHONENUMBER,
      PHONENUMSILE,
      ADDRESS,
      EMAILADDRESS )
      VALUES ( IBS-CUSTOMERNUM-INT,
      :CA-FIRST-NAME,
      :CA-LAST-NAME,
      :CA-DOB,
      :CA-HOUSE-NUM,
      :CA-HOUSE-NAME,
      :CA-POSTCODE,
      :CA-PHONE-NUM,
      :CA-PHONE-SILE,
      :CA-EMAIL-ADDRESS,
      :CA-EMAIL-ADDRESS )
      END-EXEC
    
```

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.

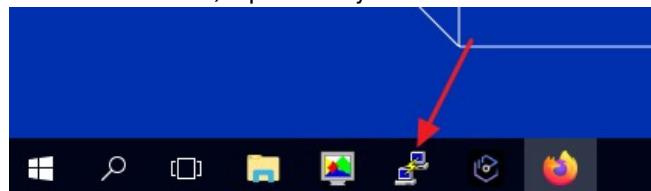
4. How to execute the Refactor phase using IBM watsonx Code Assistant for Z Refactoring Assistant

The initial installation and environment level setup will be done by the system programmer.

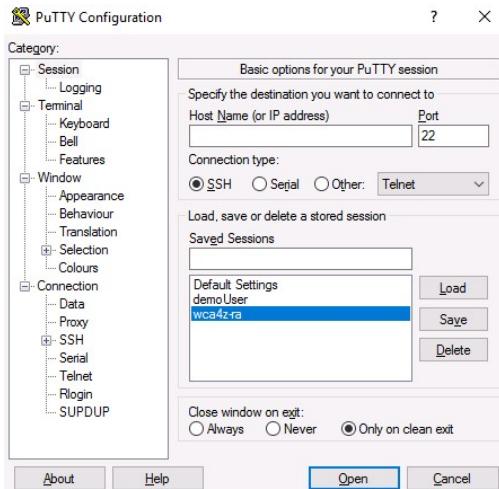
Below initial setup steps are for the user who will be using the IBM watsonx Code Assistant for Z Refactoring Assistant in zVA environment.

Procedures to access refactoring assistant on zVA.

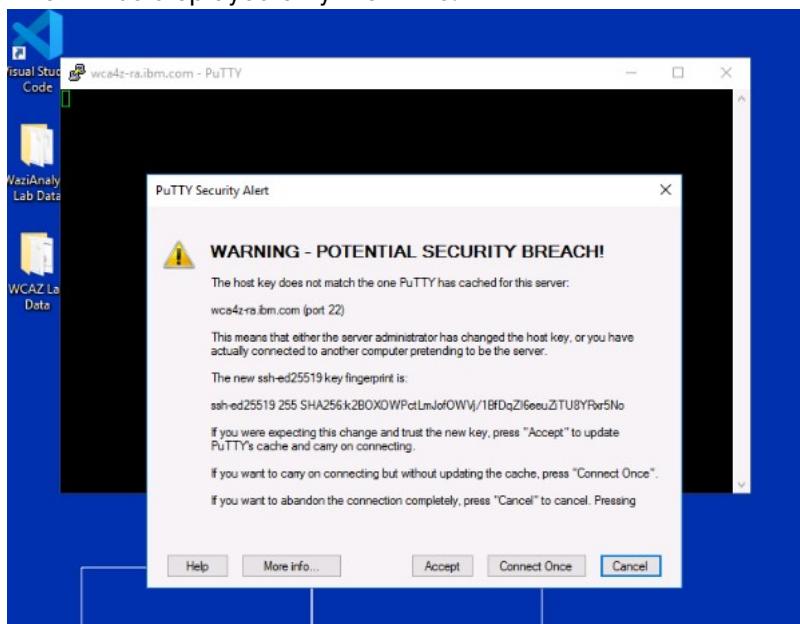
1. On RDP browser, Open Putty from taskbar.



2. select wca4z-ra and click on open.

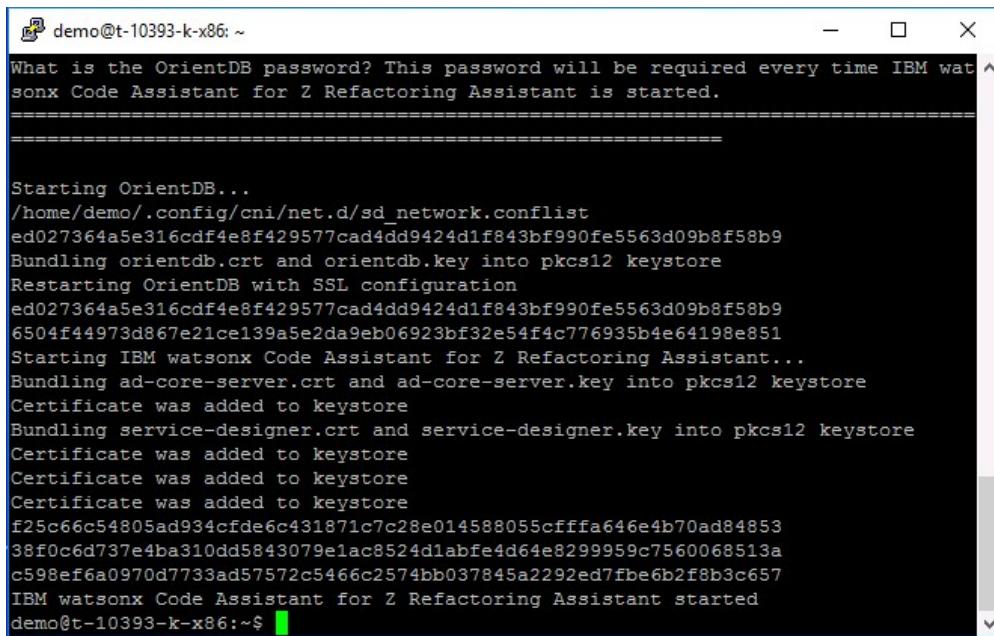


3. After clicking on Open in previous step following warning will pop-up. Select “Accept”. This will be displayed only first time.



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

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

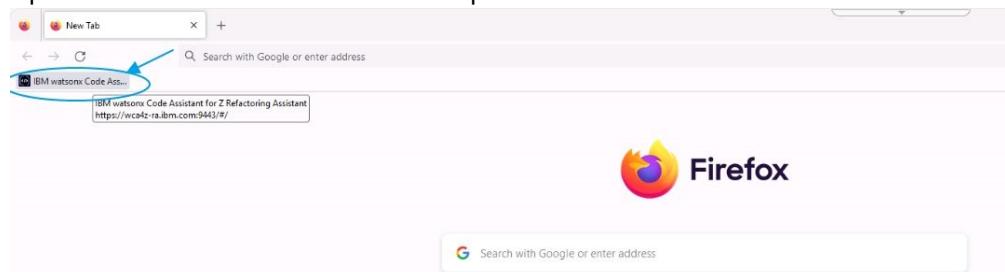


```
demo@t-10393-k-x86: ~
What is the OrientDB password? This password will be required every time IBM watsonx Code Assistant for Z Refactoring Assistant is started.
=====
=====
Starting OrientDB...
/home/demo/.config/cni/net.d/sd_network.conflict
ed027364a5e316cdf4e8f429577cad4dd9424d1f843bf990fe5563d09b8f58b9
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
38f0c6d737e4ba310dd5843079e1ac8524d1abfe4d64e8299959c7560068513a
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 watsonx Code Assistant for z.



7. Click on the Advanced tab.



Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to **wca4z-ra.ibm.com**. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

[Learn more...](#)

[Go Back \(Recommended\)](#)

[Advanced...](#)

8. Click on ‘Accept the Risk and Continue’ tab to be routed to login page.



Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to **wca4z-ra.ibm.com**. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

[Learn more...](#)

[Go Back \(Recommended\)](#)

[Advanced...](#)

wca4z-ra.ibm.com:9443 uses an invalid security certificate.

The certificate is not trusted because it is self-signed.

Error code: [MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT](#)

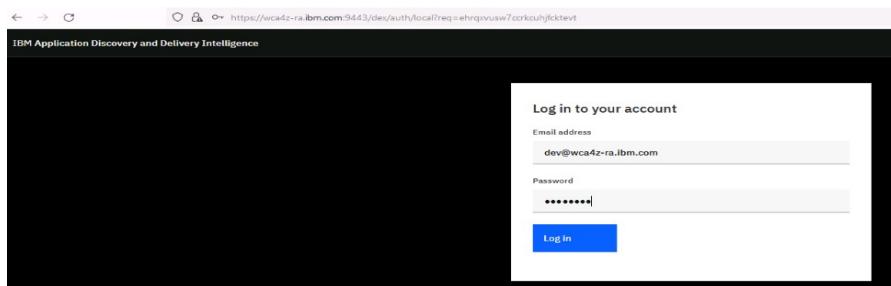
[View Certificate](#)

[Go Back \(Recommended\)](#) [Accept the Risk and Continue](#)

9. Login with following credentials

ID: dev@wca4z-ra.ibm.com

Password: password



10. After login, you can see following tabs.

- My Workspace
- 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.

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

The screenshot shows the 'Workspaces' section of the IBM Watsonx interface. At the top, there's a header bar with the application name and a 'Workspaces' tab. Below the header, a table lists workspaces under the heading 'My workspaces'. The table has columns for 'Name', 'Description', 'AD project', 'Members', and 'Actions'. A red arrow points to the 'Create workspace' button located in the top right corner of the table area.

12. In the "Create workspace" dialog

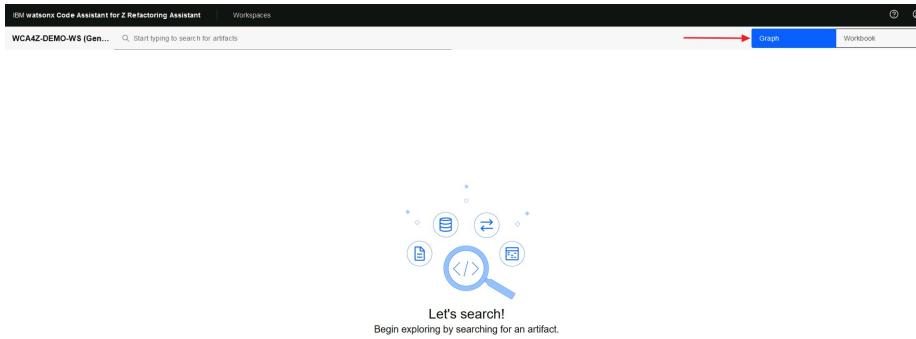
- a. Type a name for the workspace that you want to create, for example, GenApp.
 - i. Name is given as 'WCA4Z-DEM-WS' here.
- b. Provide description, e.g., workspace for WCA4Z for GenApp application.
- c. From the 'AD Project' dropdown list, select the project that you built with IBM AD Build Client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

The screenshot shows the 'Create workspace' dialog box. It has fields for 'Name' (containing 'WCA4Z-DEMO-WS'), 'Description' (containing 'workspace for WCA4Z for GenApp application'), and 'AD project' (a dropdown menu with 'GenApp' selected). At the bottom, there are 'Cancel' and 'Create' buttons, with the 'Create' button being highlighted.

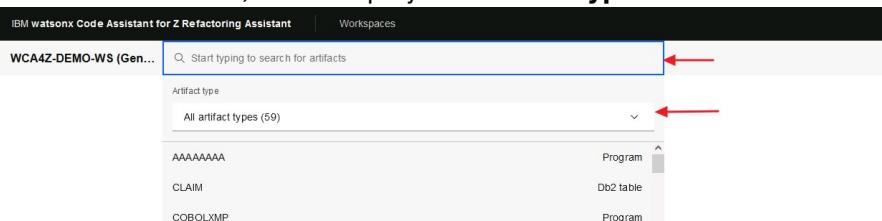
13. Click '**Create**' to create the workspace.

The screenshot shows the 'Create workspace' dialog box after the 'Create' button has been clicked. The 'Create' button is now highlighted in blue, and the rest of the dialog appears slightly dimmed.

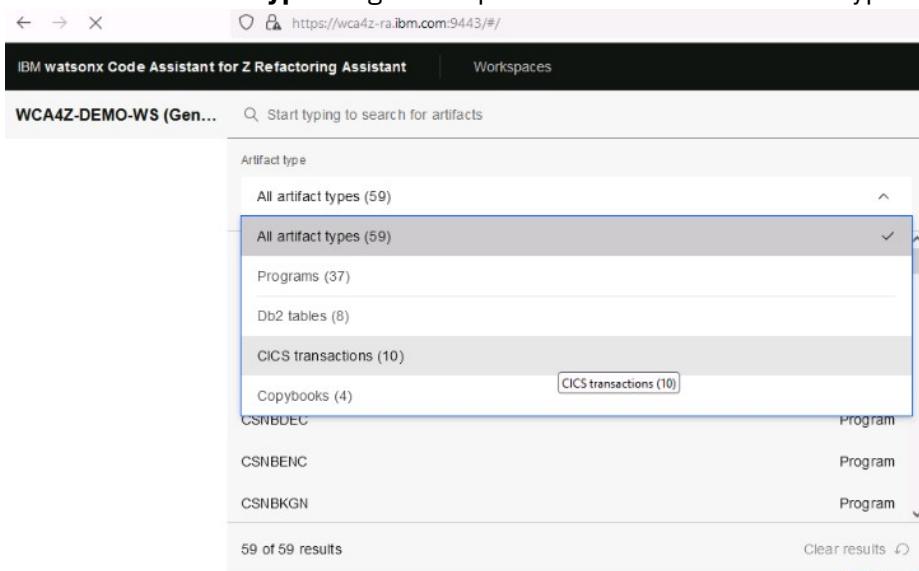
14. When the creation is completed, the Graph view of the GenApp workspace is displayed.



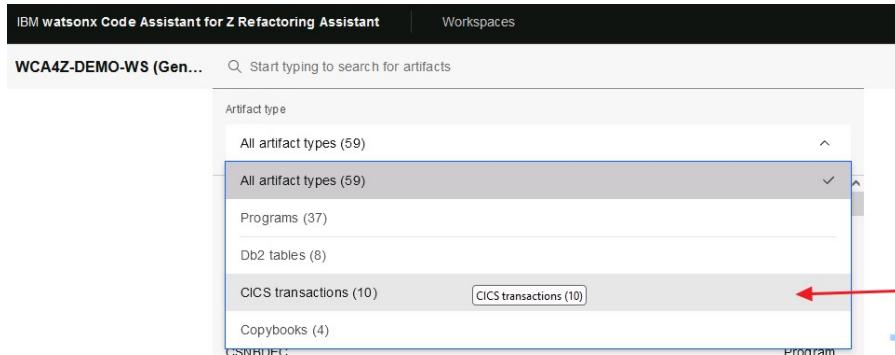
15. Click on Search bar, it will display '**all artifact types**'.



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



17. Select CICS transactions.



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

IBM Watsonx Code Assistant for Z Refactoring Assistant

Workspaces

WCA4Z-DEMO-WS (Gen...)

Start typing to search for artifacts

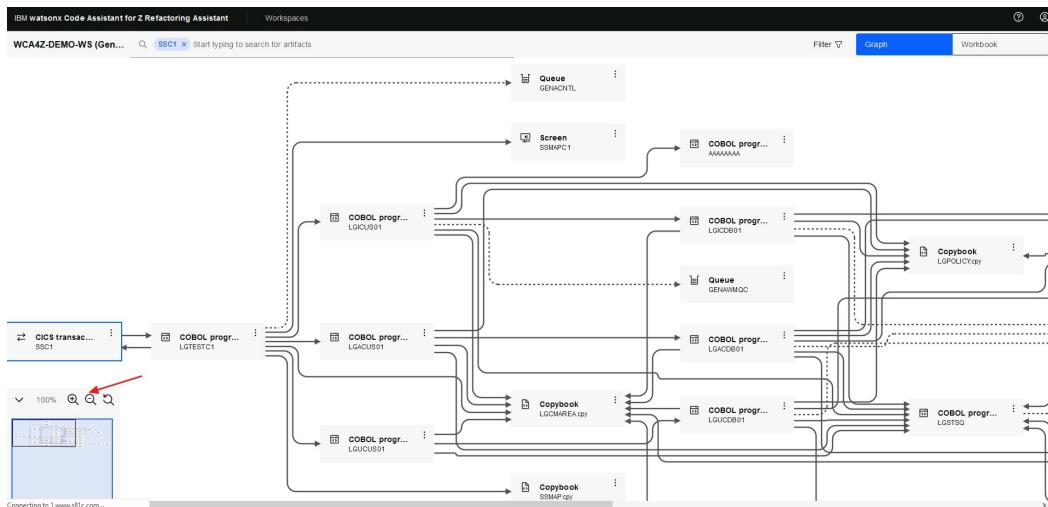
Artifact type: CICS transactions (10)

LGCF	CICS transaction
LGPF	CICS transaction
LGSE	CICS transaction
LGST	CICS transaction
SSC1	CICS transaction
SSP1	CICS transaction
SSP2	CICS transaction

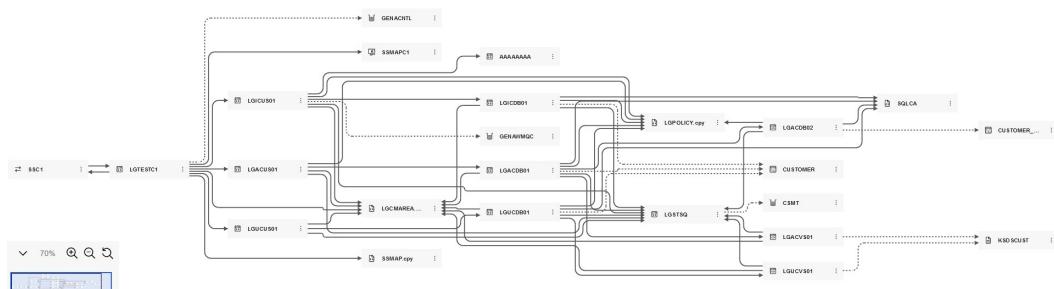
10 of 10 results

Clear results

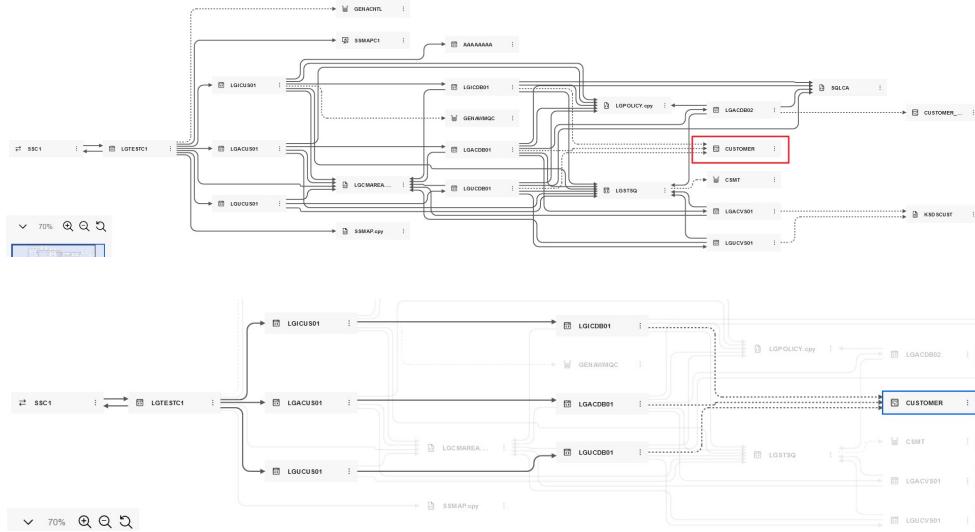
19. The graph for SSC1 transaction will open. Zoom out using zoom out icon to see complete callgraph.



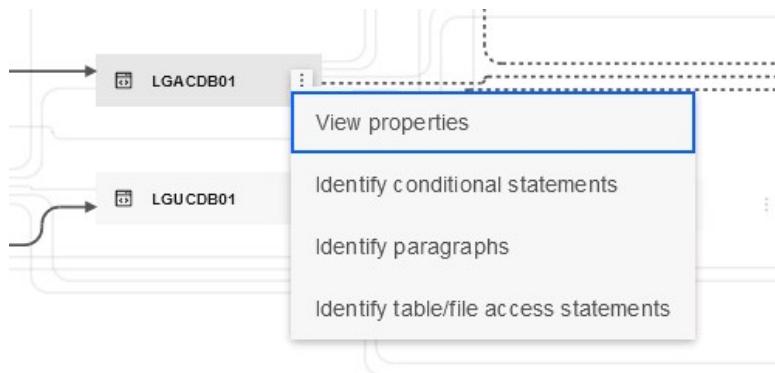
20. The graph after zooming out will be displayed.



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



22. 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.

23. 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

```

151
152
153
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-INTEGER\$.

* 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.

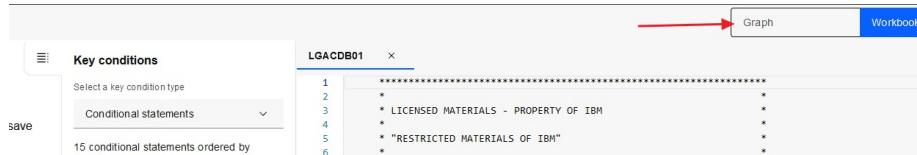
* check commarea length
ADD WS-CA-HEADER-LEN TO WS-REQUIRED-CA-LEN
ADD WS-CUSTOMER-LEN TO WS-REQUIRED-CA-LEN

* 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.

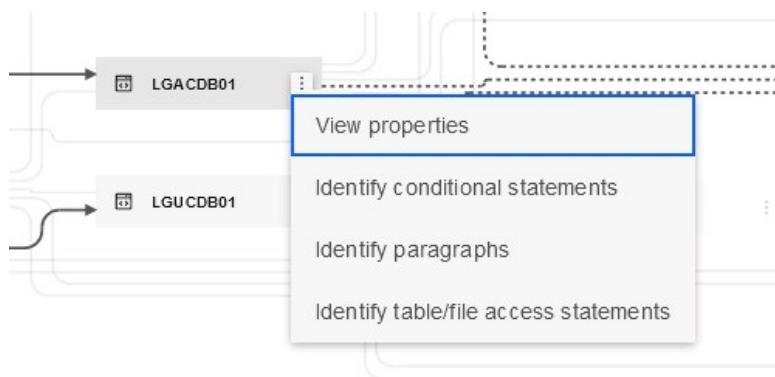
EXEC CICS LINK Program(LGACVS01)

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

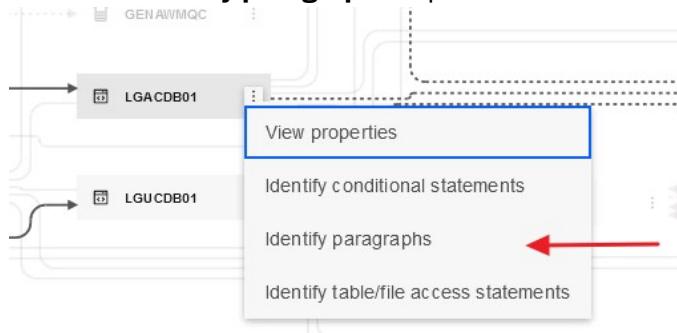


25. Click on the 3 dots/ellipses besides the LGACDB01 to get below options.

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



26. Click on “Identify paragraphs” option.



27. It will switch from graph view to Workbook view

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.

28. Click on **MAINLINE section** under Key conditions to see the paragraph in the code.

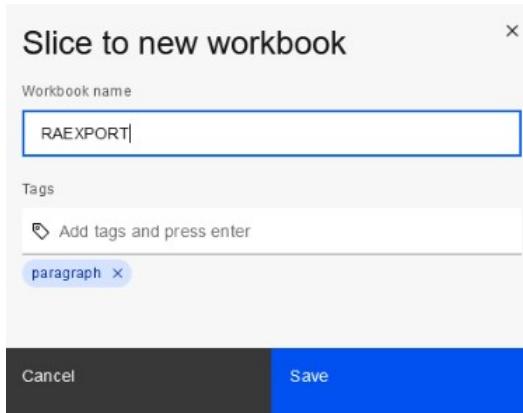
The screenshot shows the Workbench interface with the following components:

- Workbook details**: A pane on the left containing instructions for creating a workbook and a "Import workbook" button.
- Key conditions**: A pane on the right where "Paragraphs" is selected. It lists five paragraphs ordered by importance, with the first one highlighted: "MAINLINE SECTION. (line 141)".
- Code View**: The main right-hand pane displays the source code for program LGACDB01. The "MAINLINE SECTION." paragraph is highlighted in yellow.

29. Right Click on the paragraph name **MAINLINE section** appearing in the code you will get option “Slice on paragraph to new workbook”. Click on this option.

The screenshot shows the context menu for the "MAINLINE SECTION." paragraph in the code view. The "Slice on paragraph to new workbook" option is highlighted.

30. New pop-up will appear, give name as RAEXPORT, and click on Save.



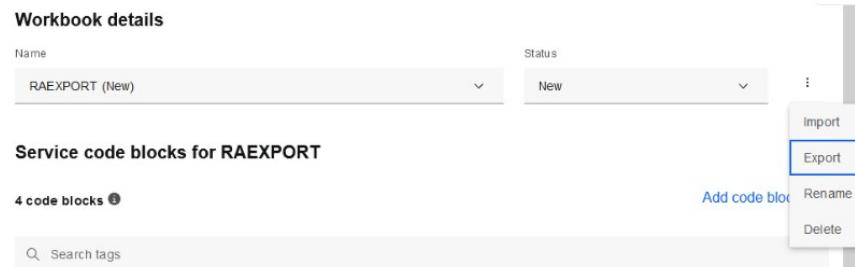
31. The code will be sliced into new workbook which will be displayed at the left side section and the selected code will be marked with dotted line on right side section.

```

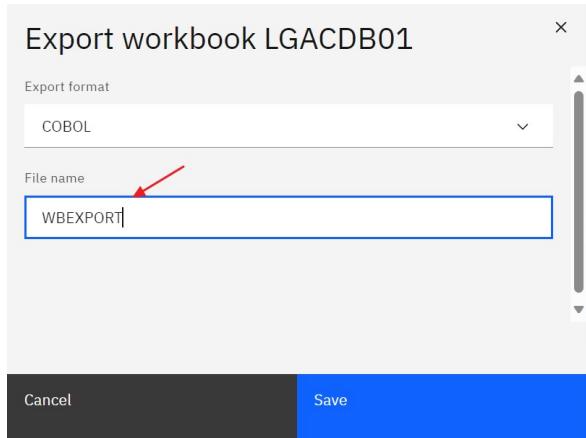
124      *-----*
125      *-----*
126      *-----*
127      *-----*
128      *-----*
129      *-----*
130      *-----*
131      *-----*
132      *-----*
133      *-----*
134      *-----*
135      *-----*
136      *-----*
137      *-----*
138      *-----*
139      *-----*
140      *-----*
141      *-----*
142      *-----*
143      *-----*
144      *-----*
145      *-----*
146      *-----*
147      *-----*
148      *-----*
149      *-----*
150      *-----*
151      *-----*
152      *-----*
153      *-----*
154      *-----*
155      *-----*
156      *-----*
157      *-----*
158      *-----*
159      *-----*
160      *-----*
161      *-----*
162      *-----*
163      *-----*
164      *-----*
165      *-----*
166      *-----*
167      *-----*
168      *-----*
169      *-----*
170      *-----*
171      *-----*

```

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

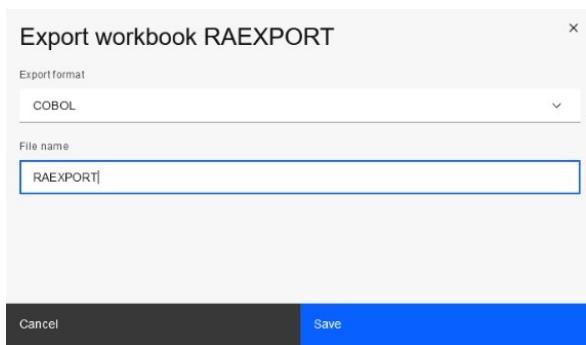


33. New pop-up will appear with pre-populated file name which is editable.



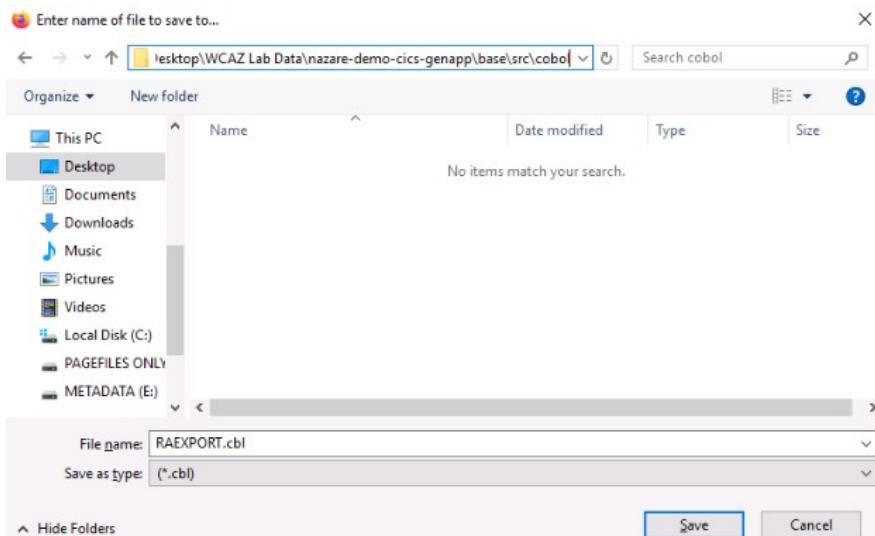
34. Change name to RAEXPORT and click on SAVE.

Please note that file name more than 8 characters is not allowed.



35. A new pop-up window will appear with default location to save the exported sliced code.

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.