



Dive into IBM i Development in VS Code

Jesse Gorzinski

Senior Business Architect

Power Systems Development

jgorzins@us.ibm.com | [@OSSJesseG](https://twitter.com/OSSJesseG)

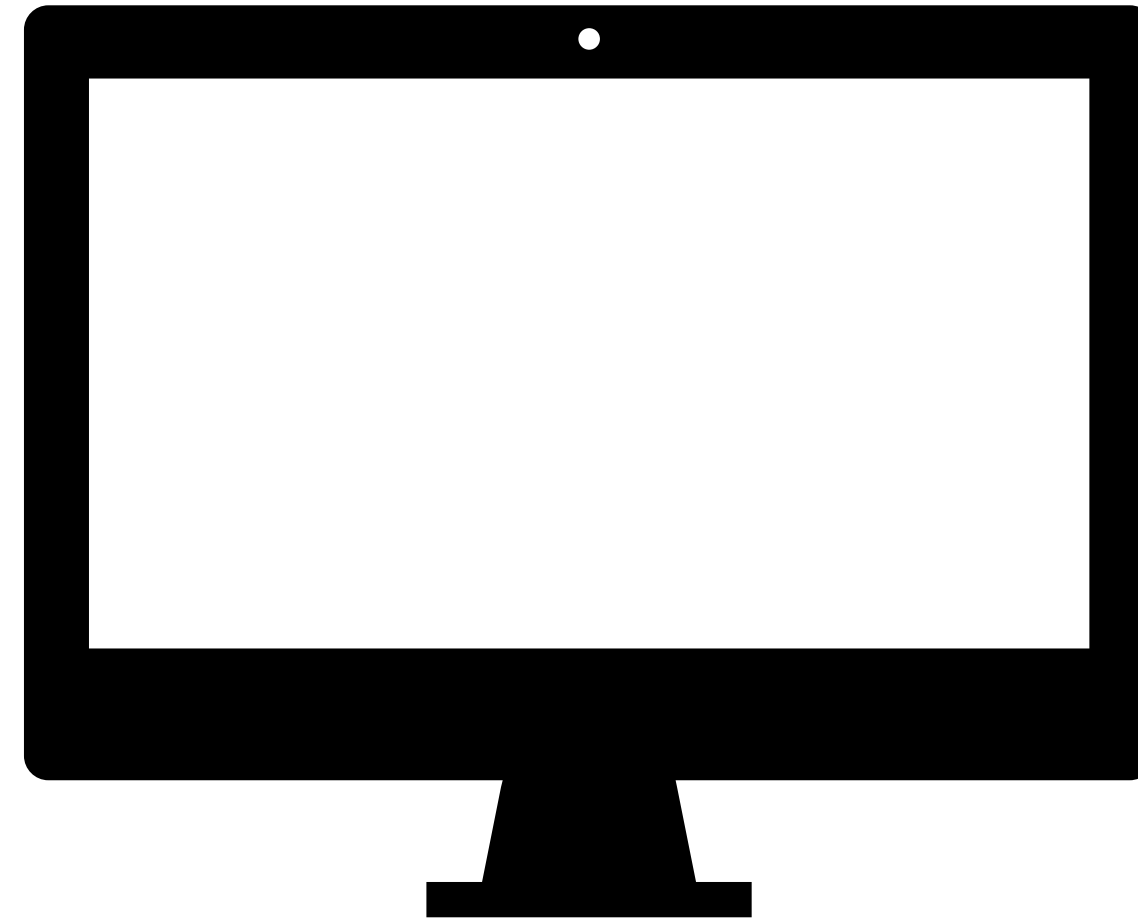
Code for IBM i

Free and open-source

50+ contributors



Mac

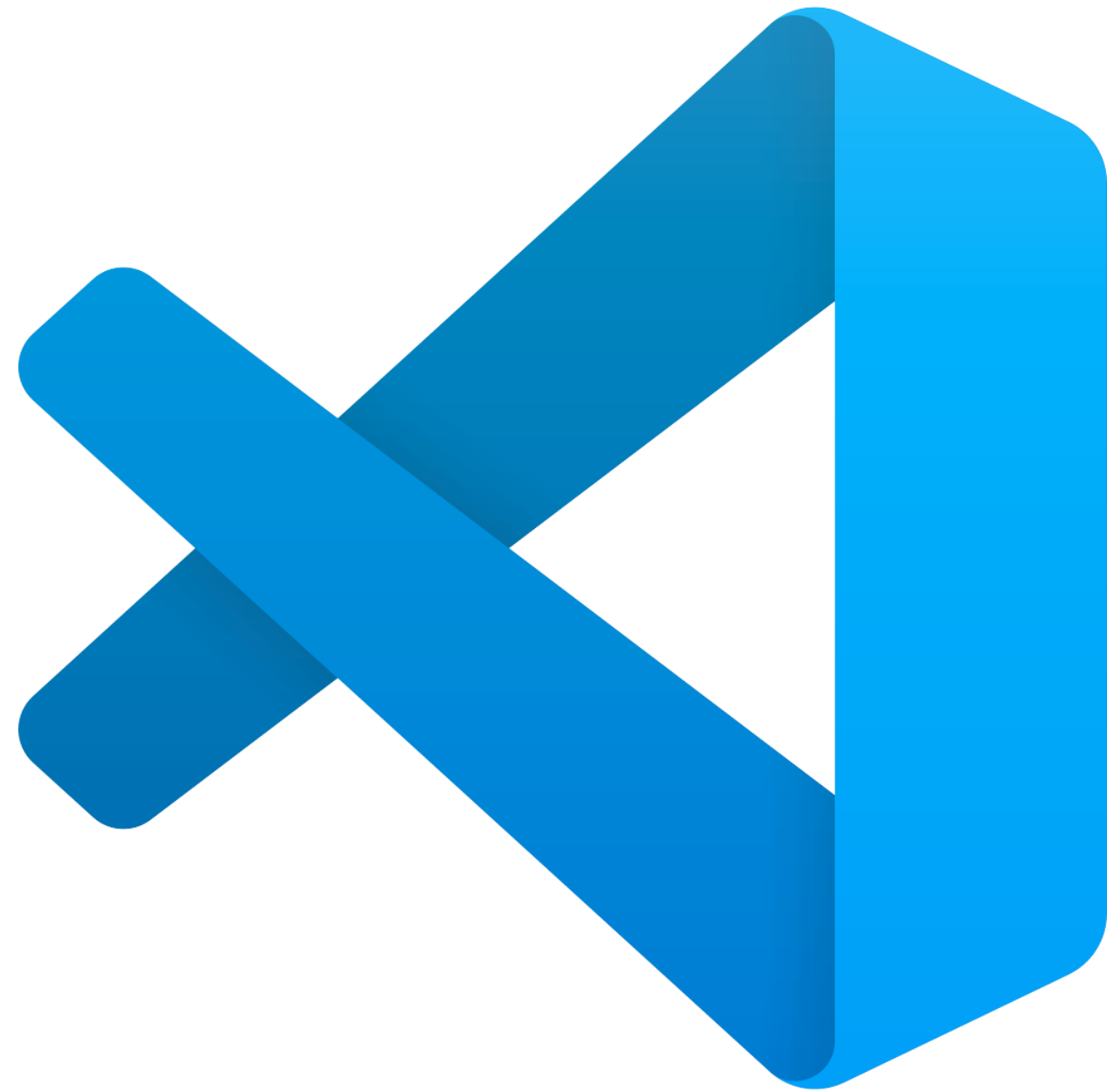


Windows

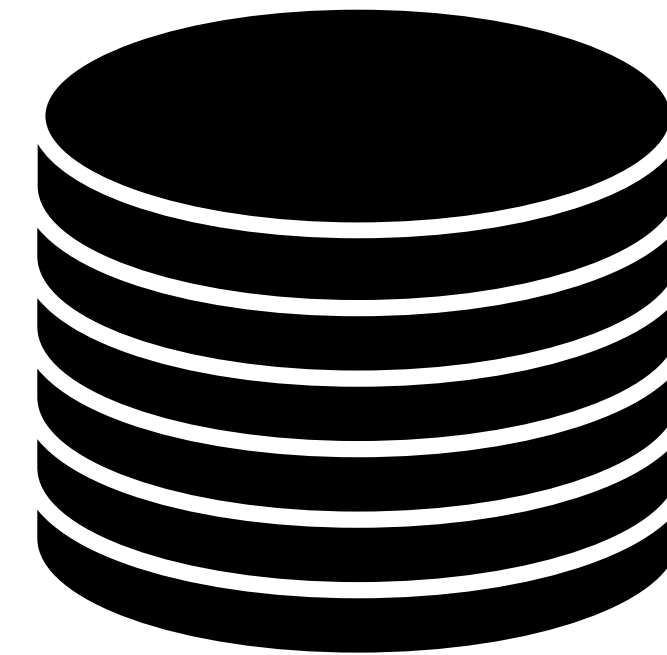


Cloud

Any platform that Visual Studio Code can run - almost all.

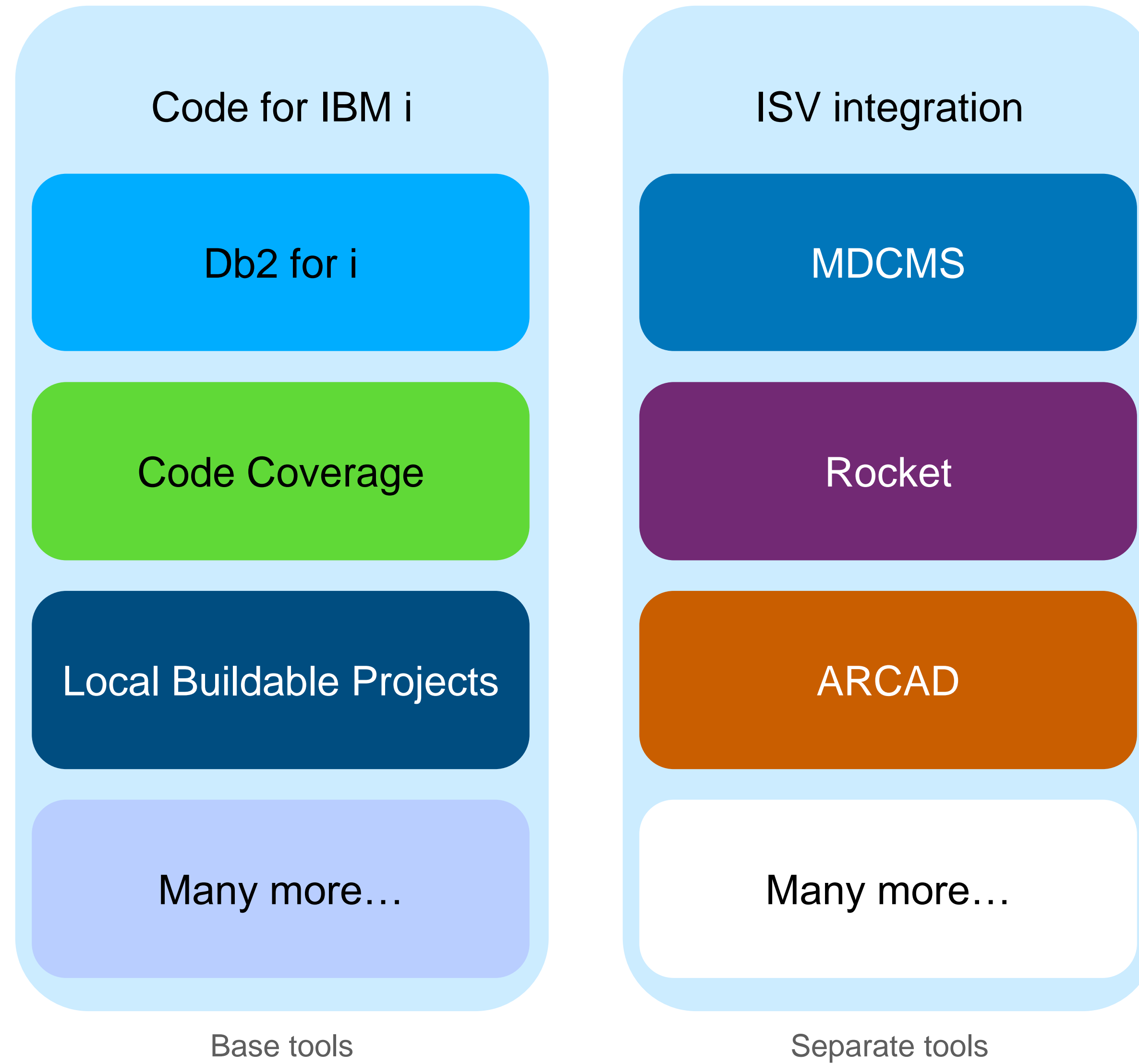


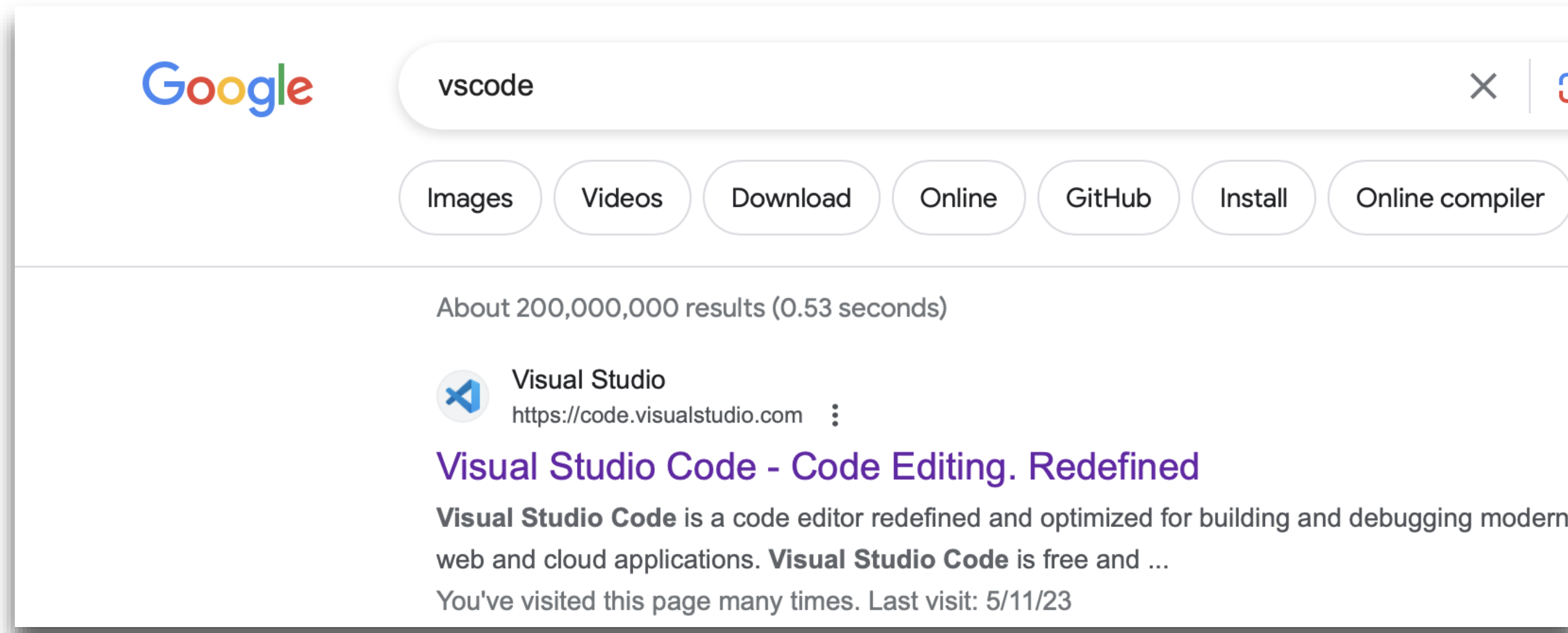
Local system



Remote IBM i

Big ecosystem






Google

vscode

Images Videos Download Online GitHub Install Online compiler

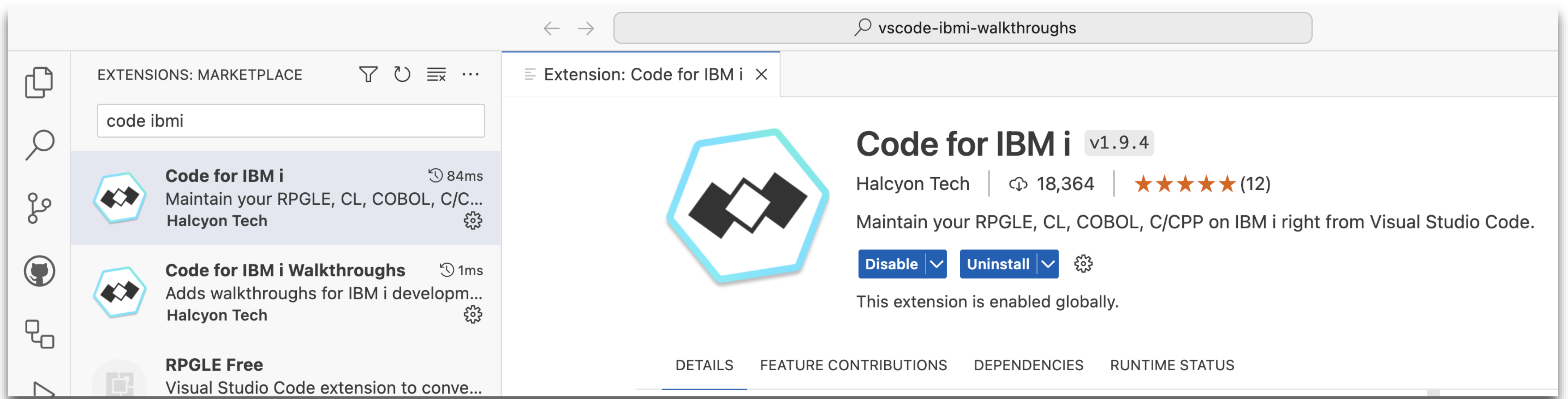
About 200,000,000 results (0.53 seconds)

 Visual Studio
https://code.visualstudio.com

Visual Studio Code - Code Editing. Redefined

Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and ...




You've visited this page many times. Last visit: 5/11/23




vscode-ibmi-walkthroughs

EXTENSIONS: MARKETPLACE

code ibmi

-  **Code for IBM i** 84ms
Maintain your RPGLE, CL, COBOL, C/C...
Halcyon Tech
-  **Code for IBM i Walkthroughs** 1ms
Adds walkthroughs for IBM i developm...
Halcyon Tech
-  **RPGLE Free**
Visual Studio Code extension to conve...

Extension: Code for IBM i



Code for IBM i v1.9.4

Halcyon Tech | 18,364 | ★★★★★ (12)

Maintain your RPGLE, CL, COBOL, C/CPP on IBM i right from Visual Studio Code.

Disable **Uninstall** ⚙️

This extension is enabled globally.

DETAILS FEATURE CONTRIBUTIONS DEPENDENCIES RUNTIME STATUS



IBM i Development Pack

Code4i | 16,010 installs | (2) | Free

Base extensions used for IBM i development in VS Code.

[Install](#)

[Trouble Installing?](#)

[Overview](#) | [Version History](#) | [Q & A](#) | [Rating & Review](#)

IBM i Development Extension Pack

This extension pack includes a set of extensions for IBM i development in Visual Studio Code:

IBM i and language specific

- [Code for IBM i](#)
- [IBMi Languages](#) (highlighting)
- [RPGLE](#)
- [COBOL](#)
- [CL](#)
- [RPGLE Free](#)
- [Code for IBM i Walkthroughs](#)
- [IBMi Notebooks](#)
- [Db2 for IBM i](#)
- [IBM i Renderer](#)
- [Error Lens](#)
- [TODO Highlight](#)

Recomendations

Extensions that are useful for IBM i but are not pack of the pack:

- [anycode](#) adds basic language tools for C/C++ and works very well with sources located on IBM i.

Notice

This pack has no affiliation with IBM.

Categories

[Extension Packs](#)

Tags

[as400](#) [cl](#) [cile](#) [cobol](#) [db2](#) [ibmi](#) [ile](#) [iseries](#)

[rpgle](#)

Works with

Universal, Web

Resources

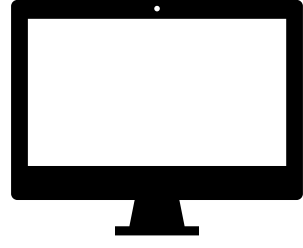
[Changelog](#)

[Download Extension](#)

More Info

Version	0.1.1
Released on	8/27/2021, 10:57:09 AM
Last updated	11/10/2023, 11:00:56 AM
Publisher	Code4i
Unique Identifier	HalcyonTechLtd.ibm-i-development-pack
Report	Report a concern





23k+ installs



1100+ daily users

FNZ Group

Amalgamated
Banks of South Africa

KS3 Solutions

Vermont
Information Processing

Directa SIM

CMA Technology Solutions

Polar Beverages


Wright Service Corp

CPB Software

Walkthroughs

F1 -> 'Open Walkthrough'

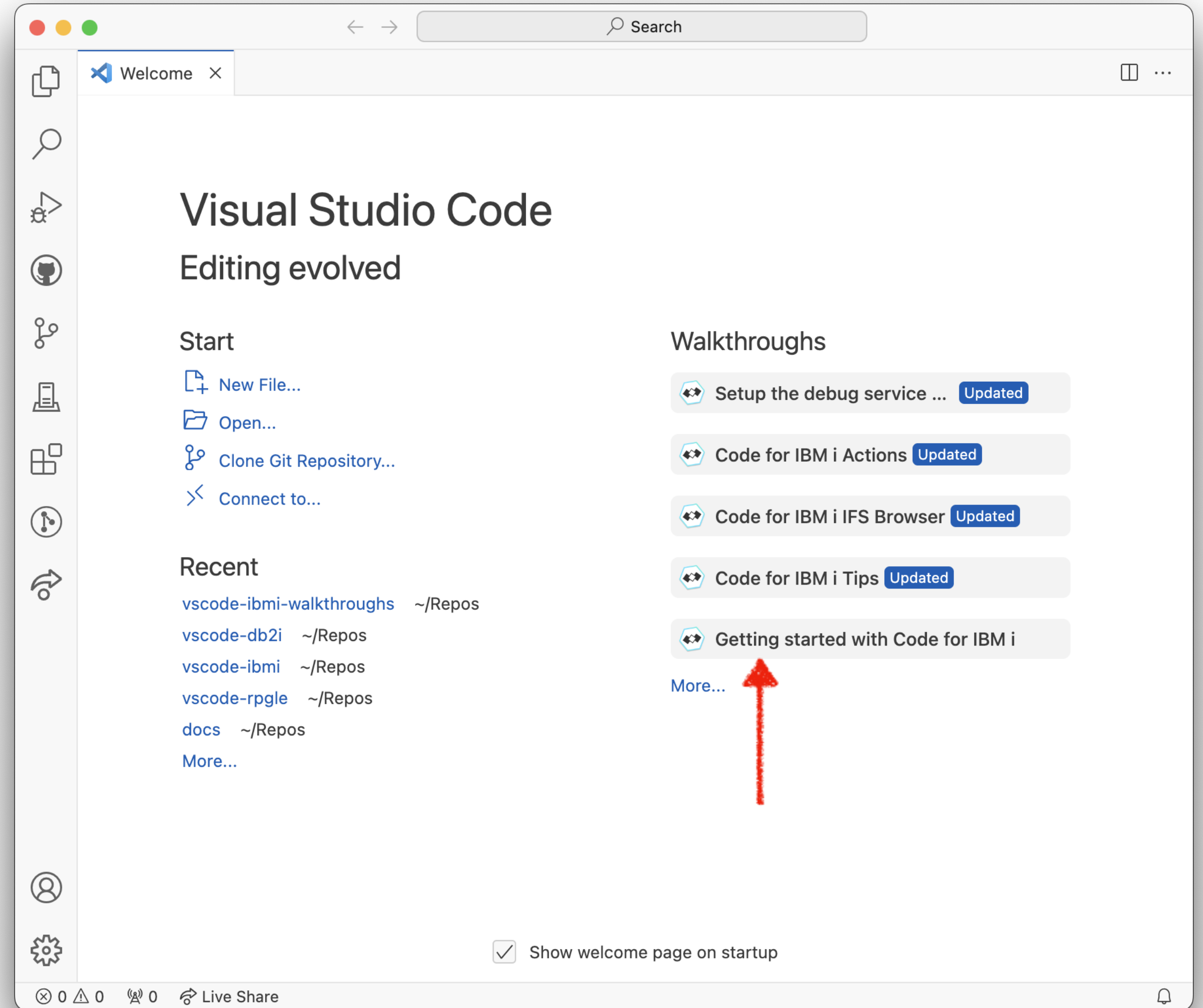
>walk

Welcome: Open **Walkthrough...** recently used 

Developer: Reset Welcome Page **Walkthrough Progress** other commands

getting

Getting started with Code for IBM i Code for IBM i Walkthroughs
Get started with what you need to write code with Code for IBM i.



Basic function

IBM I

- HELP AND SUPPORT
 - Get started
 - Open official Forum
 - Review Issues
 - Report an Issue
- PROFILES
- USER LIBRARY LIST
- OBJECT BROWSER
 - dev QDEVELOP/** (*SRCPF)
 - dev2 QBLDSYS/** (*SRCPF)
 - libs QSYS/** (*LIB)
 - qrpgle QRPGL/** (*SRCPF)
 - LIAMA LIAMA/** (*SRCPF)
 - evftempf01
 - qcbblesrc
 - qcllesrc
 - qcmdsrc
 - qddssrc
 - qrpgleref
 - qrpglesrc
 - a#abcd.rpgle
 - abcd.rpgle
 - basesrc.sqlrpgle
 - depts.sqlrpgle
 - employees.sqlrpgle 9+
 - goodbye.rpgle
 - jp.rpgle
 - noextmbr.
 - noextmbr2.
 - ocean2021.rpgle Hello ocean
 - ocean2022.rpgle Hello ocean
 - ocean2023.rpgle Hello ocean
 - report1.prtf
 - sqltools.sqlrpgle
 - srveosqls.rpgle
 - sum.rpgle
- IFS BROWSER

```

employees.sqlrpgle 9+
LIAMA > qrpglesrc > employees.sqlrpgle > ...
1  **free
2  Ctl-Opt DFTACTGRP(*no) BNDDIR('TOOLS');
3
4  Dcl-Pi EMPLOYEES;
5  DEPTNO Char(3);
6  End-Pi;
7
8  Dcl-c SQLSUCCESS '00000';
9
10 // -----*
11
12 /INCLUDE qrpgleref,constantssdfhbg
13
14 dcl-pr toUpper char(50) extproc('TOUPPER');
15 input char(50) const;
16 end-pr;
17
18 // -----*
19
20
21 Dcl-F emps WORKSTN Sfile(SFLDta:Rrn) IndDS(WkStnInd) InfDS(fileinfo);
22
23 Dcl-S Exit Ind Inz(*0ff);
24 // Hello world
25
26 Dcl-S Rrn Zoned(4:0) Inz;
27 // Hello world

```

IBM i 5250

Work with Objects

Type options, press Enter.

2=Edit authority 3=Copy 4=Delete 5=Display authority 7=Rename
8=Display description 13=Change description

Opt	Object	Type	Library	Attribute	Text
1	ABCD	*PGM	LIAMA	RPGLE	
2	DEPTS	*PGM	LIAMA	RPGLE	
3	EMPLOYEES	*PGM	LIAMA	RPGLE	
4	HANDLER	*PGM	LIAMA	CPPLE	
5	IBMIUIDSP	*PGM	LIAMA	RPGLE	
6	IBMIUIINT	*PGM	LIAMA	RPGLE	
7	MYPGM	*PGM	LIAMA	RPGLE	
8	OCEAN2021	*PGM	LIAMA	RPGLE	Hello ocean
9	OCEAN2022	*PGM	LIAMA	RPGLE	Hello ocean
10	P0	*PGM	LIAMA	RPGLE	
11	P1	*PGM	LIAMA	RPGLE	

More...

Parameters for options 5, 7 and 13 or command
====>

F3=Exit F4=Prompt F5=Refresh F9=Retrieve F11=Display names and types
F12=Cancel F16=Repeat position to F17=Position to

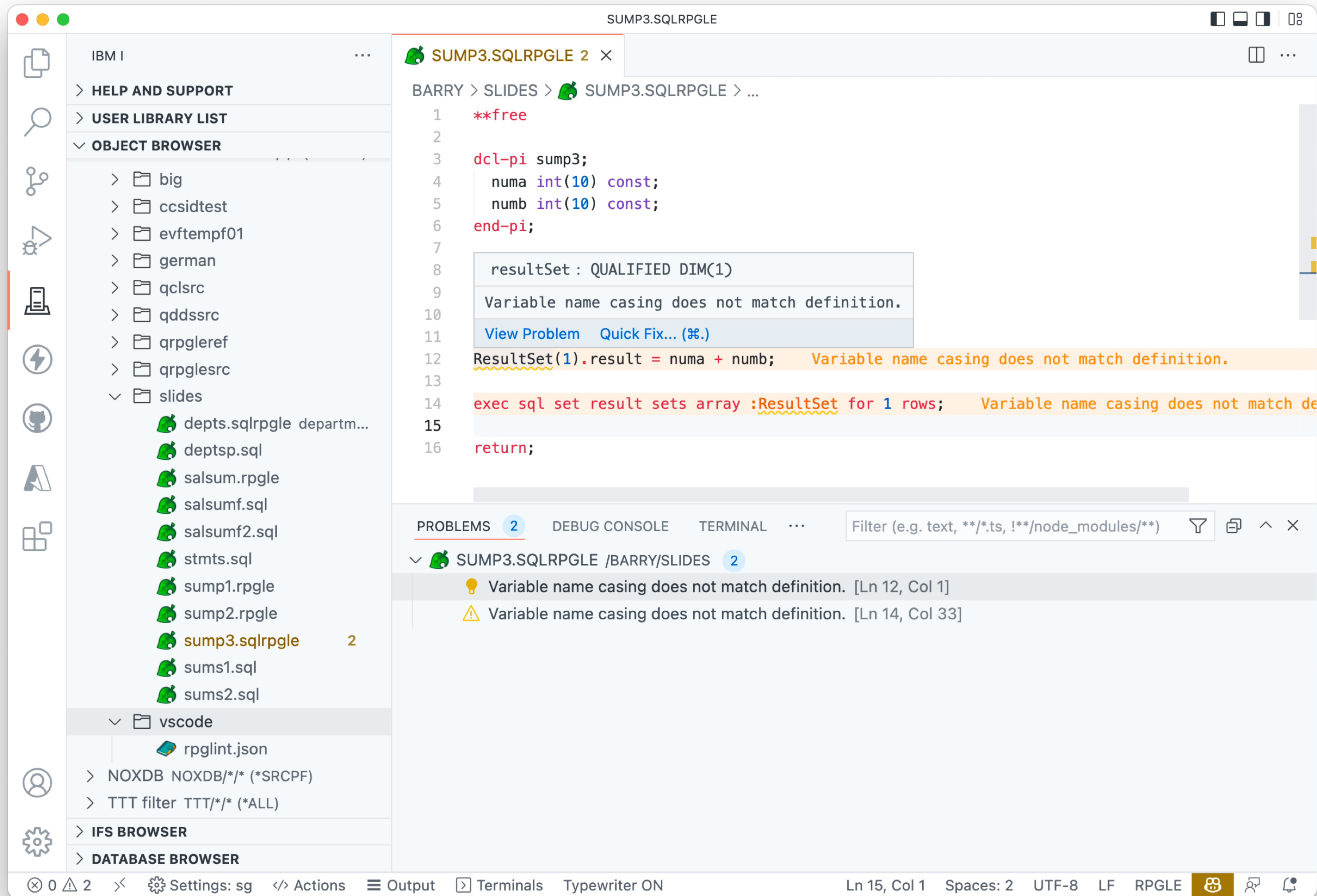
5250 002/008

PROBLEMS 36 TERMINAL OUTPUT DEBUG CONSOLE IBM I

```

*RN7001 00 04 The name of indicator is not referenced.
*RNF7089 00 1 RPG provides Separate-Indicator area for file.
*RNF7030 30 2 The name or indicator is not defined.
*RNF7503 30 2 Expression contains an operand that is not defined.
*RNF0273 40 1 Compiler not able to open the /COPY or /INCLUDE file;
directive ignored.
***** END OF MESSAGE SUMMARY *****
5770WDS V7R3M0 160422 RN IBM ILE RPG LIAMA/EMPLOYEES UT25BP18 01/23/23 16:33:58 Page 18
Final Summary
Message Totals:
Information (00) . . . . . : 65
Warning (10) . . . . . : 0
Error (20) . . . . . : 0
Severe Error (30+) . . . . . : 5
-----
Total . . . . . : 70
Source Totals:
Records . . . . . : 333
Specifications . . . . . : 244
Data records . . . . . : 0

```

The screenshot shows the Visual Studio Code interface for an IBM I project. The left sidebar contains the 'OBJECT BROWSER' with a tree view of the project structure. The main editor window displays the source code for 'SUMP3.SQLRPGLE'. The code includes a prolog, a data declaration, a SQL statement, and a return statement. Two error messages are visible in the editor, both stating 'Variable name casing does not match definition.' The bottom status bar shows the current cursor position at line 15, column 1.

OBJECT BROWSER:

- IBM I
 - HELP AND SUPPORT
 - USER LIBRARY LIST
 - OBJECT BROWSER
 - big
 - ccsidtest
 - evftempf01
 - german
 - qclsrc
 - qddssrc
 - qrppleref
 - qrpglesrc
 - slides
 - depts.sqlrpgle departm...
 - deptsp.sql
 - salsum.rpgle
 - salsumf.sql
 - salsumf2.sql
 - stmts.sql
 - sump1.rpgle
 - sump2.rpgle
 - sump3.sqlrpgle 2**
 - sums1.sql
 - sums2.sql
 - vscode
 - rpglint.json
 - NOXDB NOXDB/*/* (*SRCPF)
 - TTT filter TTT/*/* (*ALL)
 - IFS BROWSER
 - DATABASE BROWSER

Code Editor:

```
1  **free
2
3  dcl-pi sump3;
4      numa int(10) const;
5      numb int(10) const;
6  end-pi;
7
8      resultSet : QUALIFIED DIM(1)
9
10     Variable name casing does not match definition.
11     View Problem Quick Fix... (⌘.)
12     ResultSet(1).result = numa + numb; Variable name casing does not match definition.
13
14     exec sql set result sets array :ResultSet for 1 rows; Variable name casing does not match de
15
16     return;
```

PROBLEMS:

- Variable name casing does not match definition. [Ln 12, Col 1]
- Variable name casing does not match definition. [Ln 14, Col 33]

Status Bar: Ln 15, Col 1 Spaces: 2 UTF-8 LF RPGLE

employees.sqlrpgle — company_system

employees.sqlrpgle (Index) x

Users > barry > Downloads > company_system > qrpglesrc > employees.sqlrpgle

```
95      FETCH NEXT FROM empCur          96      FETCH NEXT FROM empCur
96      INTO :Employee.EMPNO,          97      INTO :Employee.EMPNO,
97      :Employee.FIRSTNME,          98      :Employee.FIRSTNME,
98      :Employee.LASTNAME,          99      :Employee.LASTNAME,
99      :Employee.JOB;              100     :Employee.JOB;
100
101     if (sqlstate = '00000');        102     if (sqlstate = '00000');
102     XID = Employee.EMPNO;          103     XID = Employee.EMPNO;
103     XNAME = %TrimR(Employee.LASTNAME) + ', ' 104     XNAME = %TrimR(Employee.LASTNAME) + ', '
104           + %TrimR(Employee.FIRSTNME) 105           + %TrimR(Employee.FIRSTNME)
105     XJOB = Employee.JOB;          106     XJOB = Employee.JOB;
106
107-   rrn += 1;                        108+   Rrn += 1;
108     Write SFLDTA;                  109     Write SFLDTA;
109   endif;                          110   endif;
110   enddo;                          111   enddo;
111
112   endif;                          112   endif;
113
114   EXEC SQL CLOSE empCur;          115   EXEC SQL CLOSE empCur;
115
116-   If (rrn > 0);                    117+   If (Rrn > 0);
117     Sfldsp = *0n;                  118     Sfldsp = *0n;
118     SFLRRN = 1;                    119     SFLRRN = 1;
119   Endif;                          120   Endif;
120   End-Proc;                        121   End-Proc;
121
122   Dcl-Proc HandleInputs;           123   Dcl-Proc HandleInputs;
123     Dcl-S SelVal Char(1);          124     Dcl-S SelVal Char(1);
124     //////////////////////////////////////////////////// 125+    Dcl-S MyTS Timestamp;
125     Dou (%EOF(emps));              126
126     ReadC SFLDTA;                  127     Dou (%EOF(emps));
127     If (%EOF(emps));              128     ReadC SFLDTA;
128       Iter;                        129     If (%EOF(emps));
129     Endif;                        130     Iter;
130
131     SelVal = %Trim(XSEL);           131     Endif;
132     //////////////////////////////////////////////////// 132
133     Select;                        133     SelVal = %Trim(XSEL);
134     When (SelVal = '5');           134+    MyTS = %timestamp;
135     DSPLY XTD;                    135
136
137     Select;                        136     Select;
138     When (SelVal = '5');           137     When (SelVal = '5');
139     DSPLY XTD;                    138     DSPLY XTD;
```

107- rrn += 1; (highlighted in red)

108+ Rrn += 1; (highlighted in green)

116- If (rrn > 0); (highlighted in red)

117+ If (Rrn > 0); (highlighted in green)

125+ Dcl-S MyTS Timestamp; (highlighted in green)

134+ MyTS = %timestamp; (highlighted in green)

Source Control: Message (%Enter to commit on 'master')

Staged Changes: 1 employees.sqlrpgle qrpglesrc 8, M

Changes: 2 actions.json .vscode M, rpglint.json .vscode U

COMMITTS, REPOSITORIES, FILE HISTORY, LINE HISTORY, BRANCHES, REMOTES, STASHES, TAGS, WORKTREES, CONTRIBUTORS, SEARCH & COMPARE

master*+ 0 14 Typewriter ON

You, 1 second ago Ln 8, Col 1 Spaces: 2 UTF-8 RPGLE

The image shows a VS Code editor window with the following components:

- Explorer:** Shows the project structure for 'IBMI-COMPANY_SYSTEM', including files like 'actions.json', 'extensions.json', 'rpglint.json', and source files like 'empfind.sqlcblle' and 'employees.pgm.sqlrpgle'.
- Outline:** Displays the structure of the COBOL program, including 'IDENTIFICATION DIVISION', 'PROGRAM-ID. BADNAME', 'DATA DIVISION', 'WORKING-STORAGE SECTION', 'LINKAGE SECTION', 'PROCEDURE DIVISION', and 'MAINLINE'.
- Code Editor:** Contains the COBOL source code for 'empfind.sqlcblle'. A lightbulb icon indicates a problem on line 2. The code includes sections for identification, data, working-storage, linkage, and procedure division.
- Problems Panel:** Shows a message: 'LNC0810: Module name 'EMPFIND' does not match outermost PROGRAM-ID name 'BADNAME'. Accepted. (10) [Ln 2, Col 22]'. The count is 105.
- Terminal:** Displays the output of the 'employees.pgm.sqlrpgle' program, showing a table of objects with columns: Opt, Object, Type, Library, Attribute, and Text.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. BADNAME.  
DATA DIVISION.  
* Called with:  
* CALL PGM(EXAMPLE3) PARM('000010' ' ' )  
WORKING-STORAGE SECTION.  
EXEC SQL INCLUDE SQLCA END-EXEC.  
* Variables  
LINKAGE SECTION.  
* Input struct  
01 base.  
05 empno PIC X(6).  
* Output struct  
01 result.  
05 fullname PIC X(25).  
PROCEDURE DIVISION USING base result.  
MAINLINE.  
MOVE " " TO fullname.  
IF (EMPNO of BASE IS NOT = " ")  
EXEC SQL  
SELECT FIRSTNME || " " || LASTNAME  
INTO :fullname  
FROM SAMPLE.EMPLOYEE  
WHERE EMPNO = :empno  
END-EXEC  
END-IF.
```

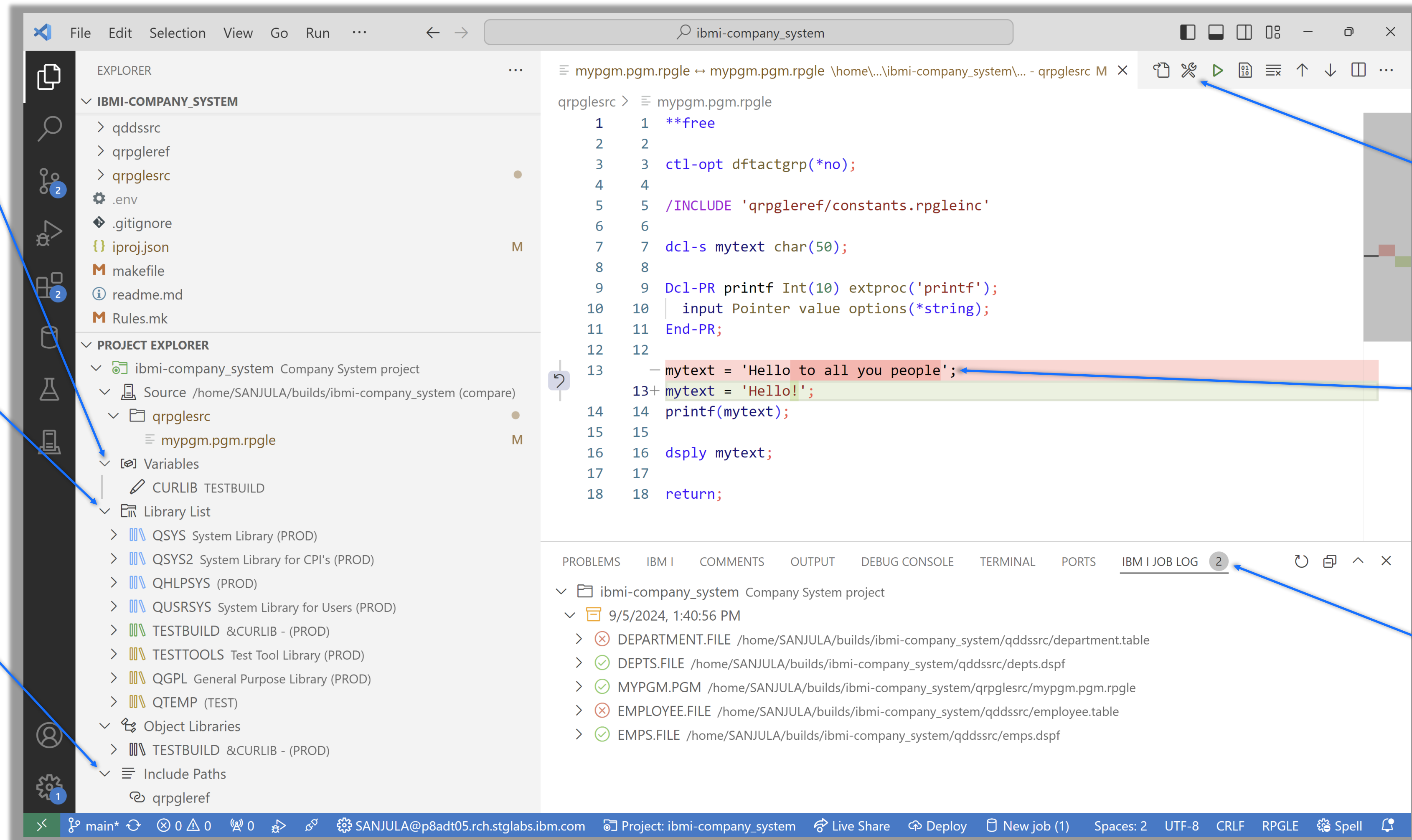
Opt	Object	Type	Library	Attribute	Text
—	EMP	*FILE	SAMPLE	DDMF	
—	EMP_ACT	*FILE	SAMPLE	DDMF	
—	EMP_PHOTO	*FILE	SAMPLE	PF	
—	EMP_P00001	*FILE	SAMPLE	LF	
—	EMP_RESUME	*FILE	SAMPLE	PF	
—	EMPACT	*FILE	SAMPLE	DDMF	
—	EMPLOYEE	*FILE	SAMPLE	PF	
—	EMPPROJACT	*FILE	SAMPLE	PF	
—	EMPFILE	*PGM	LIAMA	CBLLLE	
—	EMPFIND	*PGM	LIAMA	CBLLLE	
—	EMPLOYEEES	*PGM	LIAMA	RPGLLE	

```
5250          MW          002/016  
007/020
```

Project Explorer

Overview

The ultimate tool for local development on IBM i!



Set variables

Manage library list

Modify include paths

Build and Compile

Local source vs. IFS source

View job logs

Migrate Source from QSYS

CVTSRCPF
from BOB



QSYS members in source
physical files
↓
Properly encoded,
terminated, and named
source files in an IFS
directory

↓
Download to local project

↓
Rename extensions
↓
Convert includes/copy
directives to Unix style
paths

Source Orbit



File Edit Selection View Go ...

Untitled (Workspace)

IBM I

HELP AND SUPPORT

- Get started
- Open official Forum
- Review Issues
- Report an Issue
- Download Logs

USER LIBRARY LIST

OBJECT BROWSER

Filter 1 QGPL/*.* (*LIB)

qtpl.lib General Purpose Library (PROD)

Migrate Source (QGPL)

Convert all members in source physical files of a library to IFS files and then download them to a workspace folder. To learn more about [cvtsrcpf](#), click [here](#).

Settings Clean Up Source Files

Source Library

The name of the library containing the source files to migrate.

QGPL

Default CCSID

The CCSID to be used when the source file is 65535.

*JOB

Workspace folder

The workspace folder to which the files are to be downloaded to.

my-project

Import Member Text

Imports member text at the top of the source as comments.

Lowercase Filenames

The generated source file names will be in lowercase.

Assign to Variable

Migrate Source

New Source File...

Change Description...

Copy...

Rename...

Delete... Delete

Run Action... Ctrl+E

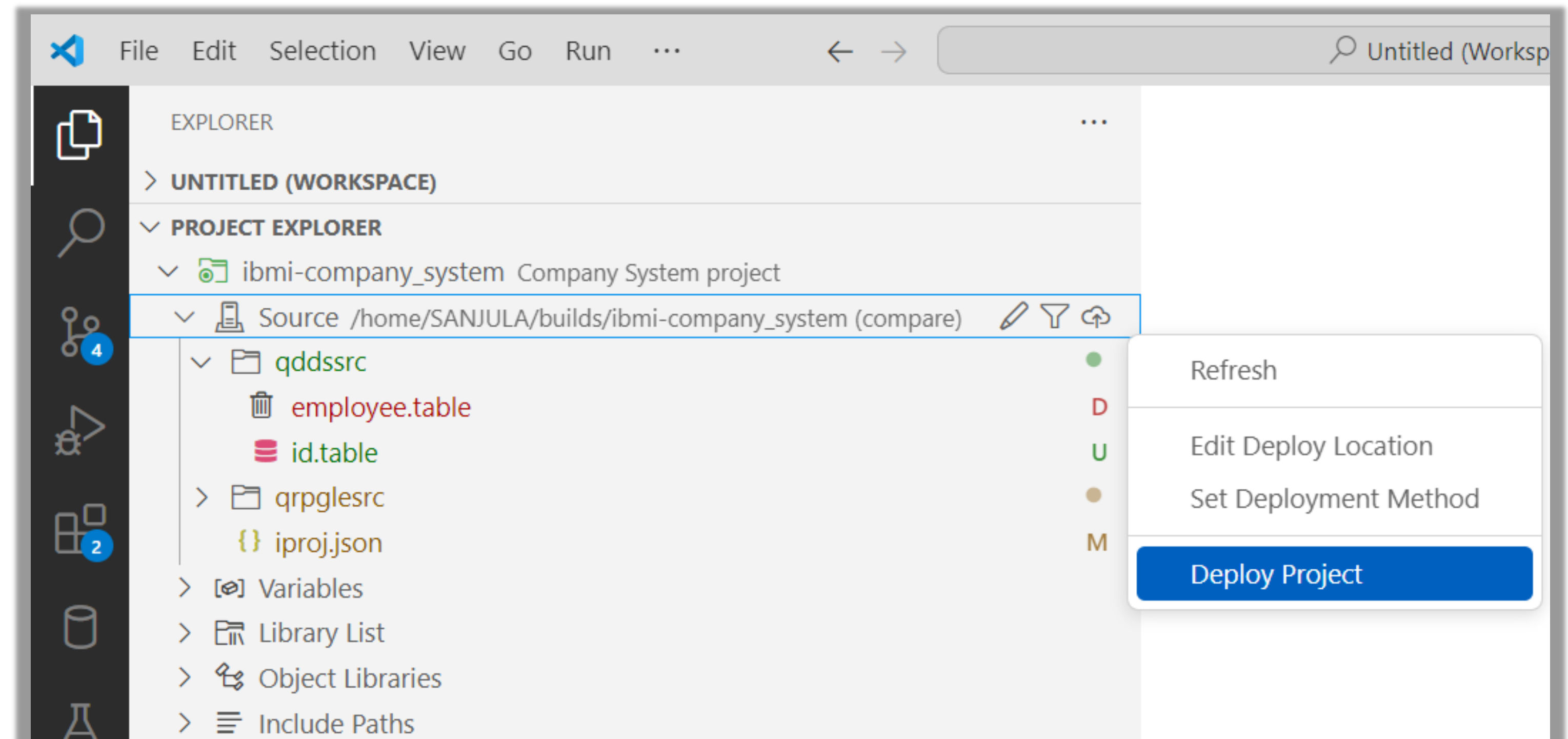
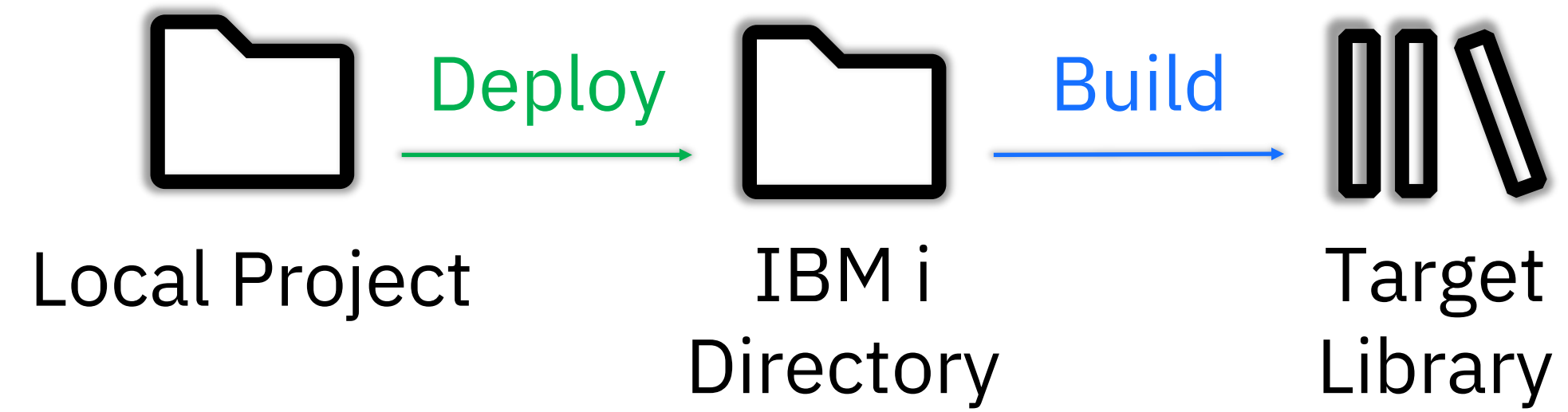
Refresh

IFS BROWSER

main 0 0 0 SANJULA@i7adt03.rch.stglabs.ibm.com Project: my-project Live Share Deploy New job (1) Spell

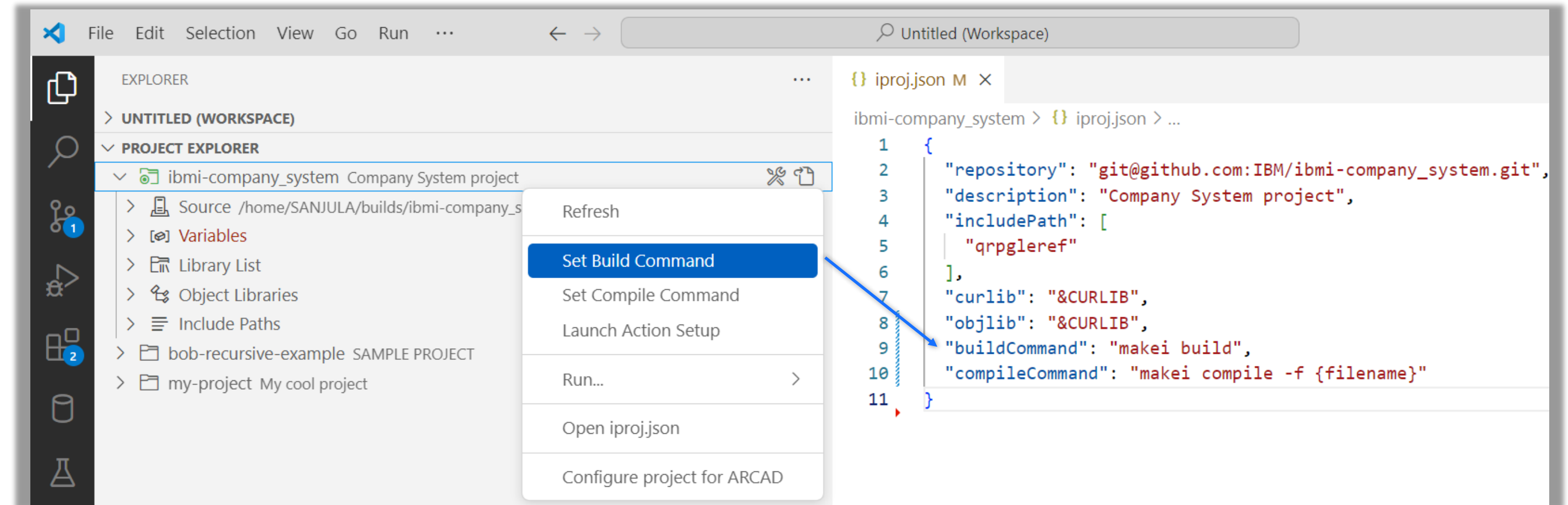
Deployment

- Set deploy location
 - Where source gets uploaded to
 - Typically set one
 - Each developer gets a unique location
 - Each repository gets a unique location
- Set deployment method
 - Compare (typically the safest)
 - Changes (typically the fastest)
 - Working Changes
 - Staged Changes
 - All
- Deploy project
 - Moves files to deploy location based on deployment method

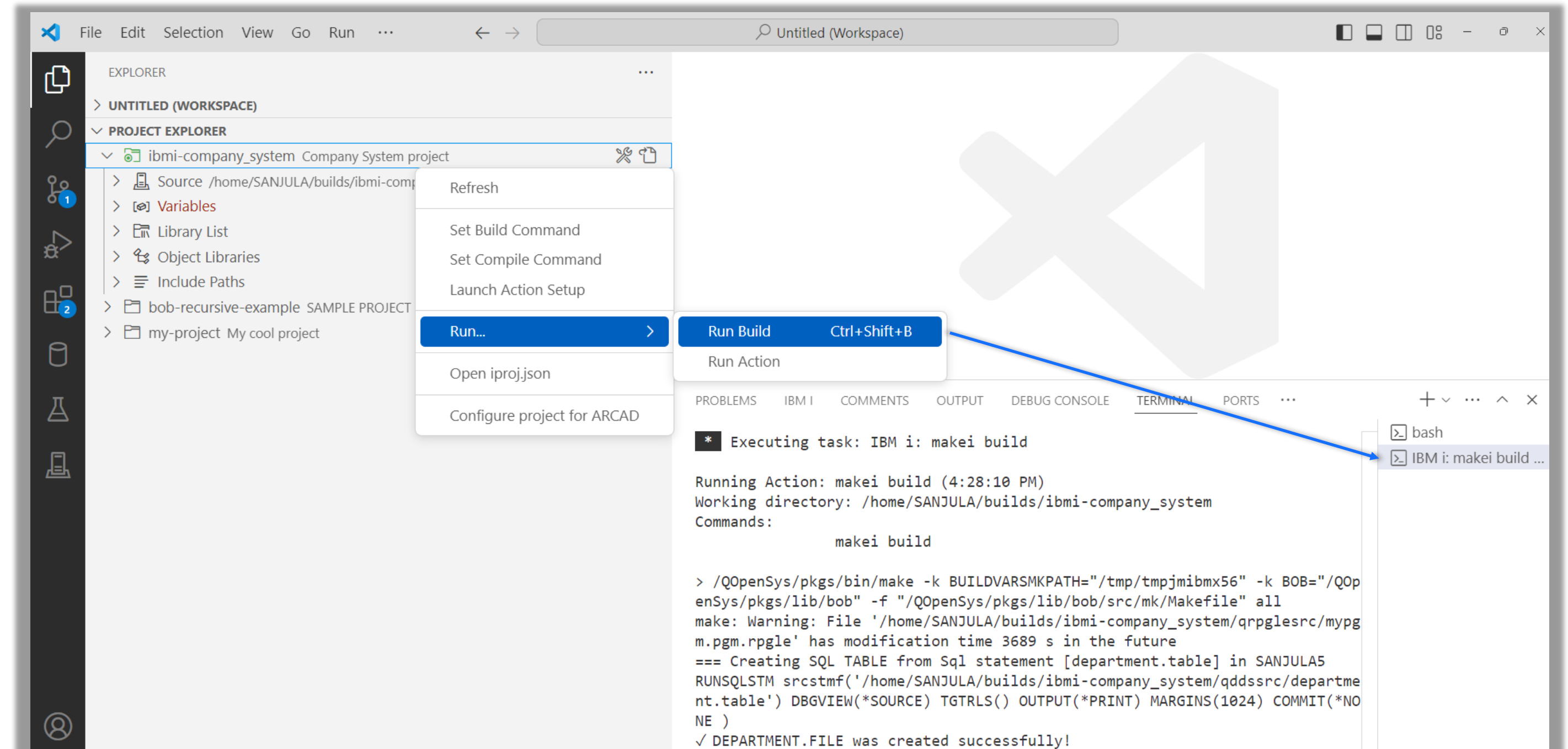


Build and Compile

- 1 Deploy
- 2 Run build or compile command
(any build framework)
- 3 Download logs and evfevent files

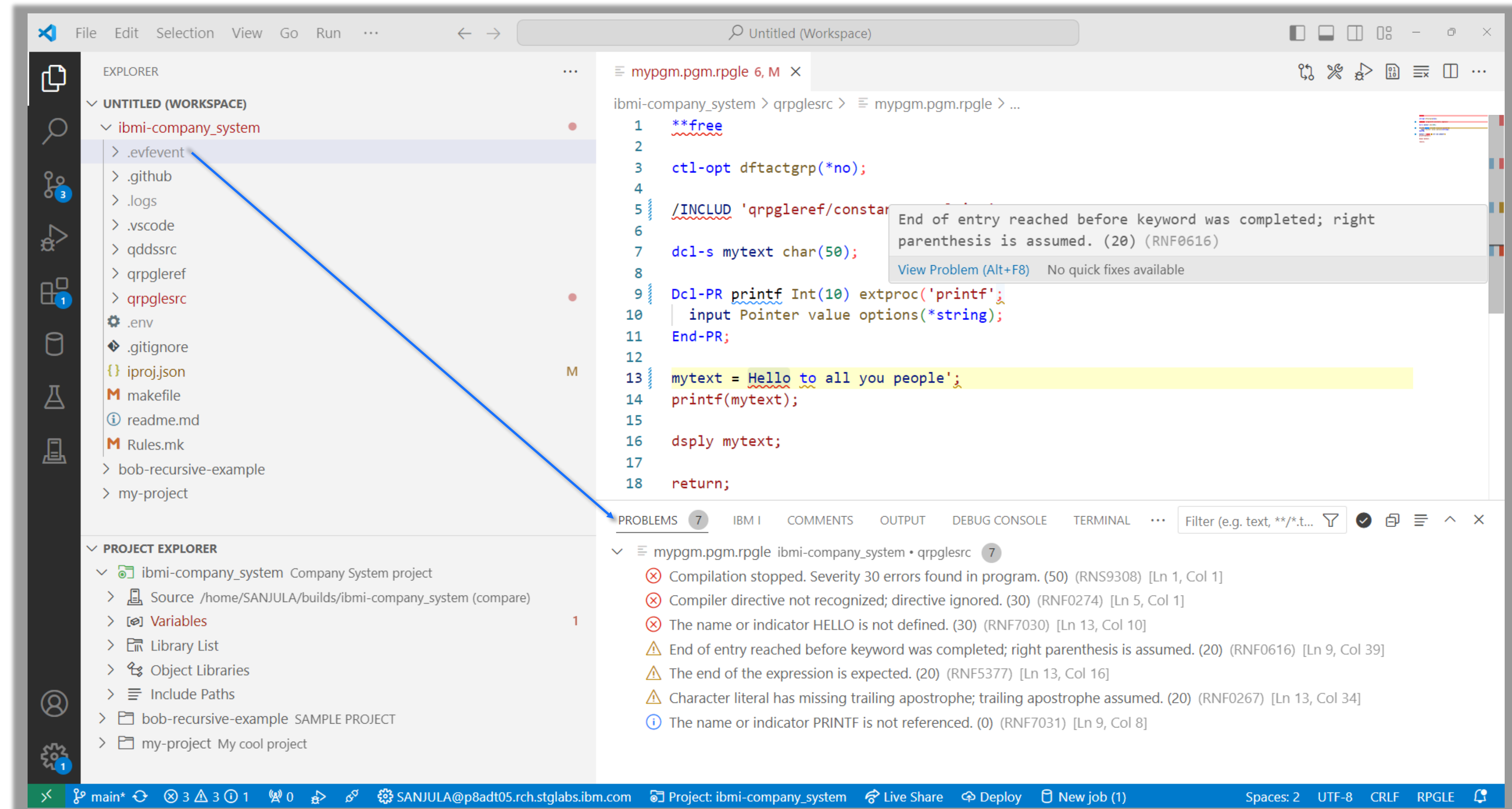


- Building
 - Set build command
 - Run Build (*Ctrl+Shift+b* or *Cmd+Shift+b*)
- Compiling
 - Set compile command
 - Run compile (*Ctrl+Shift+c* or *Cmd+Shift+c*)
 - On active editor
 - On file or directory in File Explorer
 - On file or directory in Source



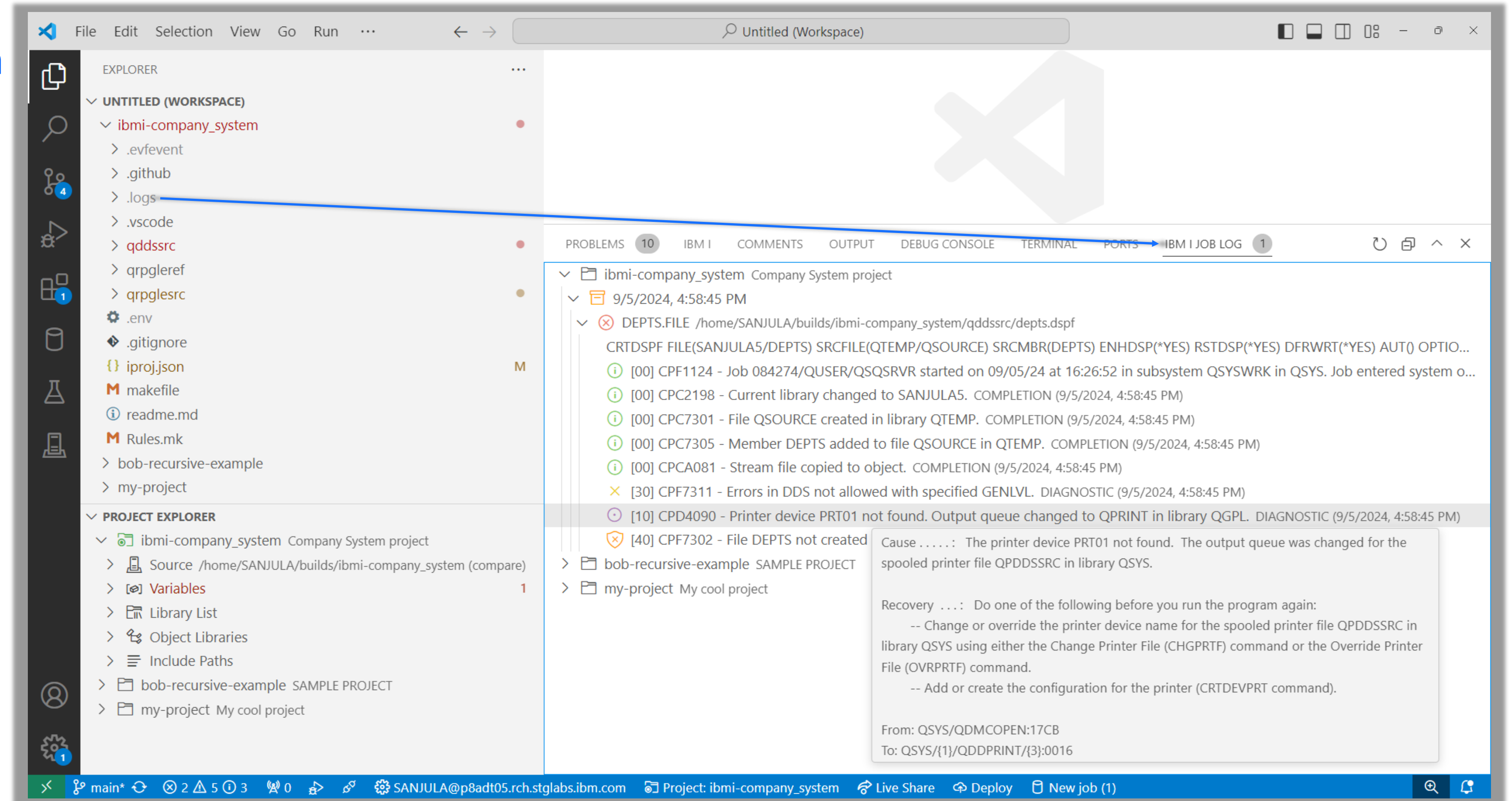
View Diagnostics

- Visualize build or compile diagnostics in the Problems view
- Evfevent file diagnostics are dumped in a .evfevent directory after a build or compile
- Diagnostics are also rendered inline in the source file



View Job Logs

- Visualize and manage anything that could be seen in an IBM i job log including second level help
- Job log and spool files are dumped in .logs directory after a build or compile
- Track up to 10 of the previous logs in memory
- Organized by the ILE objects being built
- Filter by failed objects or severity



Source Orbit

Source Orbit

What does it do?

- It's a CLI tool
- Build dependency lists for RPGLE, DDS, SQL, CL, etc
- Generate impact analysis information
- Create scripts to automate builds of your application code
- Cleans up your repository
- It's open-source!

```
ics — zsh — 111x30
barry@Liams-MBP ics % so -h

-d <dir>
--cwd <dir>      To see the directory of where source code lives.
                  The default is the current working directory.

-f <relative paths>
--files <relative paths>
                  this option will still scan the entire working
                  directory, but will single out these specific files
                  when generating other files. If no build file is
                  specified, dependencies will be printed.

-bf make|bob|imd|json  Create build files of a specific format
                       Example: -bf make

-bl <name>          Set the BRANCHLIB environment variable based on
                   a user provided branch name, and will write it out.
                   Example: -bl feature/123-cool-idea
                           -bl bug/123-bad-move

Options for project cleanup:

-ar                Run the auto-rename process after scanning all code
                   Ensure it is run inside of source control.
                   Rename program sources to have the '.pgm.' attribute in the name
                   Rename RPGLE copybooks found (based on include statements) to be '.rpgleinc'

-fi                Fix includes for RPGLE sources
                   Ensure it is run inside of source control.
```

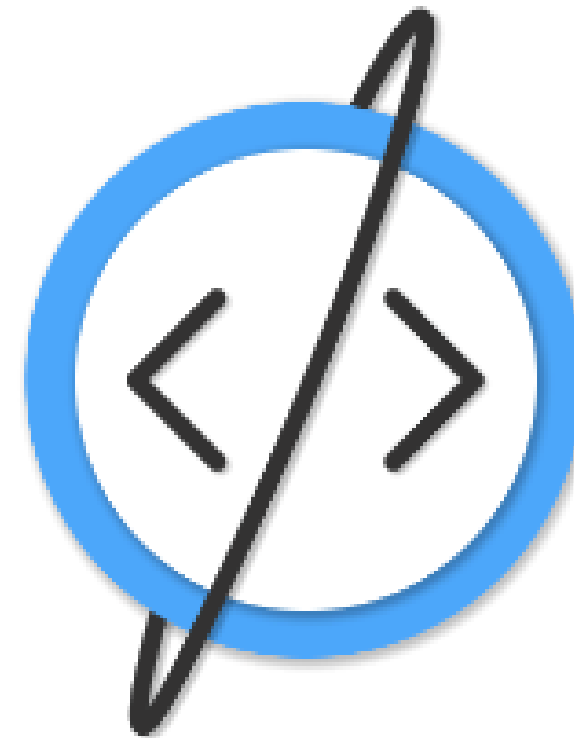
Source Orbit

Source Orbit is a dependency management tool. As IBM i developers start using Git for their RPGLE, CL, DDS and SQL, we want to provide them with excellent tools to help them understand their source code. Source orbit is available for use as both a VS Code extension and CLI tool.

-  [Install from VS Code Marketplace](#)
-  [Download from Open VSX Registry](#)
-  [Install CLI from NPM](#)
-  [View Documentation](#)
-  [See Releases](#)

Extension **v1.0.2** installs **60**

CLI **v1.0.2** downloads **405/month**



Install

```
> npm i @ibm/sourceorbit
```

Repository

 github.com/IBM/sourceorbit

Homepage

 ibm.github.io/sourceorbit/#/

Weekly Downloads

31

Version

1.0.2

License

Apache 2

Unpacked Size

166 kB

Total Files

20

<https://www.npmjs.com/package/@ibm/sourceorbit>

Repository cleanup

The screenshot shows a Visual Studio Code editor with a diff view. The left pane shows the Explorer view with a project structure for 'BBS400'. The main editor shows a diff between the local 'Working Tree' and a repository. The diff highlights several lines where copy statements have been corrected. The original code in the working tree (left) has paths like 'H/COPY DVBBS400/CURRENTSRC,CBKOPTIMIZ' and 'D/COPY DVBBS400/CURRENTSRC,CBKDTAARA'. The repository version (right) has corrected paths: '/copy 'src/CBKOPTIMIZ.rpgleinc'' and '/copy 'src/CBKDTAARA.rpgleinc''. Other lines in the diff show corrections to 'D wCfgKey S 6A' to '/copy 'src/CBKPCFGDCL.rpgleinc''. The status bar at the bottom indicates the repository version is 'You, 1 second ago' and the current line is 'Ln 10, Col 1'.

Show code in a repository where the copy statements have been corrected

Impact Analysis

worksofliam / ibmi-company_system-rmake

Code Pull requests 1 Actions Projects Wiki Security Insights Settings

← Impact analysis

Update utils.sqlrpgle #1 Re-run all jobs

Summary

Jobs

- Code scan (18.x)

Run details

- Usage
- Workflow file

Triggered via pull request 2 hours ago

Job	Status	Total duration	Artifacts
worksofliam opened #2 test/utils_impact	Success	22s	-

impact.yaml
on: pull_request

Matrix: Code scan

- 1 job completed

Show all jobs

Code scan (18.x) summary

Impact Analysis

Touched objects:

- UTILS.MODULE : qrpglesrc/utils.sqlrpgle

UTILS.MODULE

- UTILS.MODULE (qrpglesrc/utils.sqlrpgle)
 - UTILS.SRVPGM (qsrvsrc/utils.bnd)
 - DEPTS.PGM (qrpglesrc/depts.pgm.sqlrpgle)
 - \$(APP_BNDDIR).BNDDIR (no source)

Summary

Jobs

- Code scan (18.x)

Run details

- Usage
- Workflow file

Code scan (18.x) summary

Impact Analysis

Touched objects:

- BANKING.MODULE : qrpglesrc/banking.sqlrpgle
- DEPTS.PGM : qrpglesrc/depts.pgm.sqlrpgle
- UTILS.MODULE : qrpglesrc/utils.sqlrpgle
- UTILS.SRVPGM : qsrvsrc/utils.bnd

BANKING.MODULE

- BANKING.MODULE (qrpglesrc/banking.sqlrpgle)
 - BANKING.SRVPGM (no source)
 - \$(APP_BNDDIR).BNDDIR (no source)

DEPTS.PGM

- DEPTS.PGM (qrpglesrc/depts.pgm.sqlrpgle)

Changes to this object have no impact.

UTILS.MODULE

- UTILS.MODULE (qrpglesrc/utils.sqlrpgle)
 - UTILS.SRVPGM (qsrvsrc/utils.bnd)
 - DEPTS.PGM (qrpglesrc/depts.pgm.sqlrpgle)
 - \$(APP_BNDDIR).BNDDIR (no source)

pr.yaml

name: Source Orbit Impact Report

on:

pull_request:

types: [opened]

jobs:

so-impact:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

with:

fetch-depth: 0

- uses: actions/setup-node@v3

with:

node-version: 18

- run: npm i -g @ibm/sourceorbit

- run: npm i -g @ibm/ibmi-ci

- name: Generate impact information

run: so -bf imd -l `git diff --name-only origin/main origin/\${GITHUB_HEAD_REF}`

- name: Adding markdown

run: cat impact.md >> \$GITHUB_STEP_SUMMARY

Impact Analysis

worksofliam / ibmi-company_system-rmake

Code Pull requests 1 Actions Projects Wiki Security Insights Settings

← Impact analysis

Update utils.sqlrpgle #1 Re-run all jobs

Summary

Jobs

- Code scan (18.x)

Run details

- Usage
- Workflow file

Triggered via pull request 2 hours ago

Job	Status	Total duration	Artifacts
worksofliam opened #2 test/utils_impact	Success	22s	-

impact.yaml
on: pull_request

Matrix: Code scan

- 1 job completed

Show all jobs

Code scan (18.x) summary

Impact Analysis

Touched objects:

- UTILS.MODULE : qrpglesrc/utils.sqlrpgle

UTILS.MODULE

- UTILS.MODULE (qrpglesrc/utils.sqlrpgle)
 - UTILS.SRVPGM (qsrvsrc/utils.bnd)
 - DEPTS.PGM (qrpglesrc/depts.pgm.sqlrpgle)
 - \$(APP_BNDDIR).BNDDIR (no source)

Summary

Jobs

- Code scan (18.x)

Run details

- Usage
- Workflow file

Code scan (18.x) summary

Impact Analysis

Touched objects:

- BANKING.MODULE : qrpglesrc/banking.sqlrpgle
- DEPTS.PGM : qrpglesrc/depts.pgm.sqlrpgle
- UTILS.MODULE : qrpglesrc/utils.sqlrpgle
- UTILS.SRVPGM : qsrvsrc/utils.bnd

BANKING.MODULE

- BANKING.MODULE (qrpglesrc/banking.sqlrpgle)
 - BANKING.SRVPGM (no source)
 - \$(APP_BNDDIR).BNDDIR (no source)

DEPTS.PGM

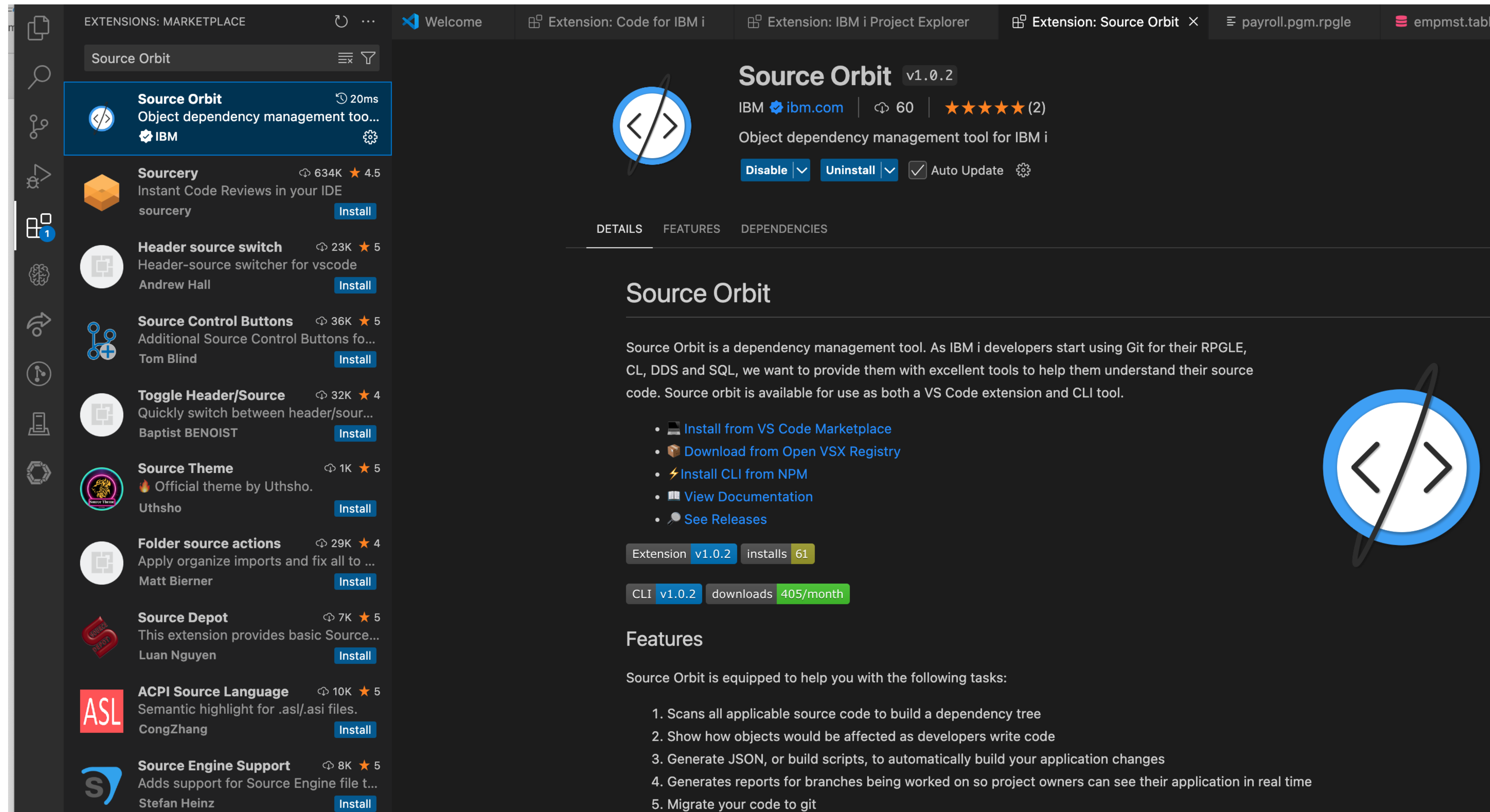
- DEPTS.PGM (qrpglesrc/depts.pgm.sqlrpgle)

Changes to this object have no impact.

UTILS.MODULE

- UTILS.MODULE (qrpglesrc/utils.sqlrpgle)
 - UTILS.SRVPGM (qsrvsrc/utils.bnd)
 - DEPTS.PGM (qrpglesrc/depts.pgm.sqlrpgle)
 - \$(APP_BNDDIR).BNDDIR (no source)

Source Orbit as VS Code Extension



The screenshot displays the VS Code Extensions Marketplace interface. On the left, a list of extensions is shown, with 'Source Orbit' highlighted. The main panel shows the details for the 'Source Orbit' extension by IBM. The extension is version 1.0.2, has 60 installs, and a 5-star rating. It is described as an 'Object dependency management tool for IBM i'. The interface includes tabs for 'DETAILS', 'FEATURES', and 'DEPENDENCIES'. The 'DETAILS' tab is active, showing a description of the tool and a list of installation options: 'Install from VS Code Marketplace', 'Download from Open VSX Registry', 'Install CLI from NPM', 'View Documentation', and 'See Releases'. Below this, it shows 'Extension v1.0.2' with 'installs 61' and 'CLI v1.0.2' with 'downloads 405/month'. The 'FEATURES' tab is also visible, listing five tasks that Source Orbit helps with: scanning source code for a dependency tree, showing affected objects, generating JSON/build scripts, generating reports for branches, and migrating code to git. A large blue circular logo with a slash and angle brackets is overlaid on the right side of the screenshot.

Source Orbit v1.0.2
IBM [ibm.com](#) | 60 | ★★★★★ (2)
Object dependency management tool for IBM i

[Disable](#) [Uninstall](#) Auto Update

DETAILS FEATURES DEPENDENCIES

Source Orbit

Source Orbit is a dependency management tool. As IBM i developers start using Git for their RPGLE, CL, DDS and SQL, we want to provide them with excellent tools to help them understand their source code. Source orbit is available for use as both a VS Code extension and CLI tool.

- [Install from VS Code Marketplace](#)
- [Download from Open VSX Registry](#)
- [Install CLI from NPM](#)
- [View Documentation](#)
- [See Releases](#)

Extension **v1.0.2** installs **61**

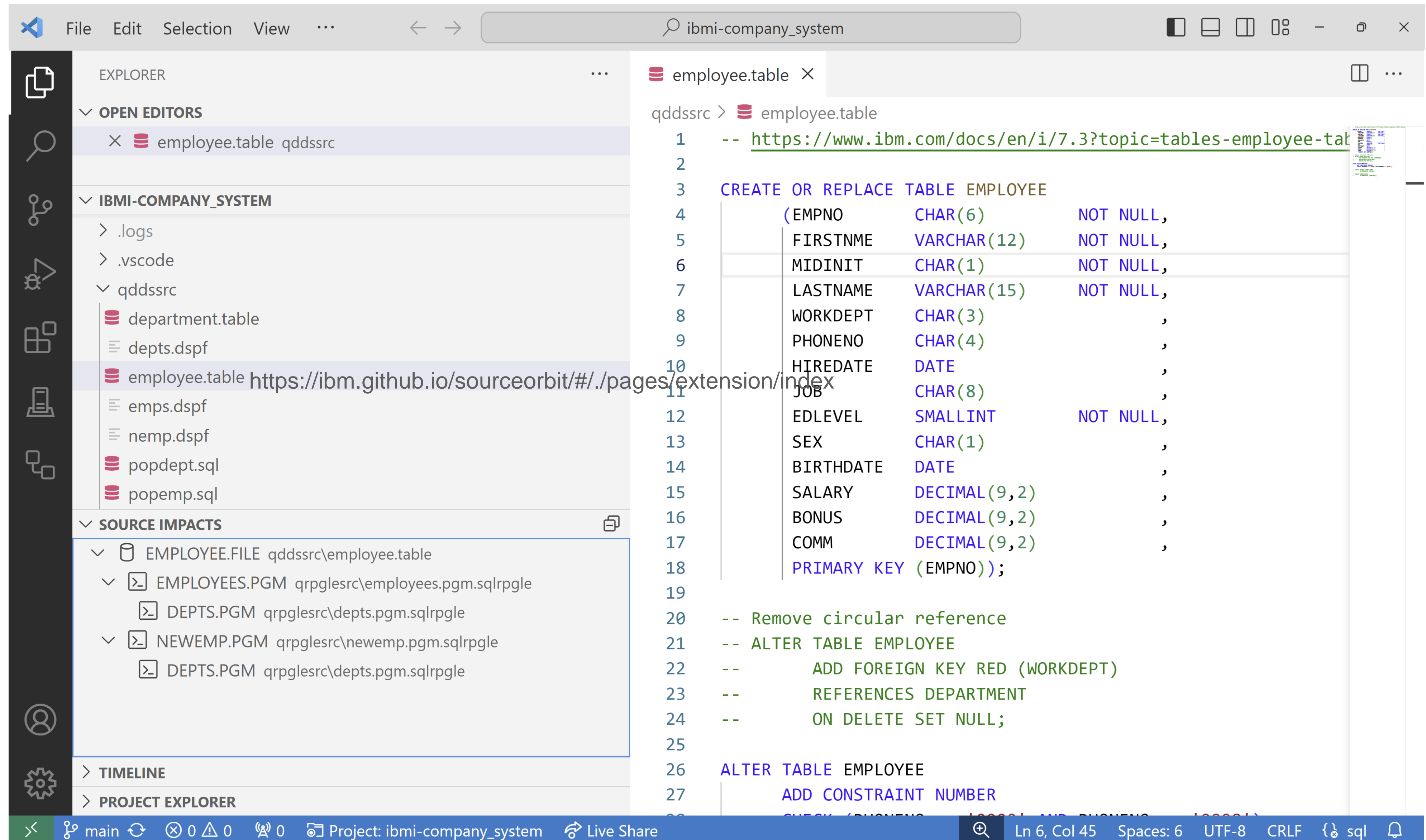
CLI **v1.0.2** downloads **405/month**

Features

Source Orbit is equipped to help you with the following tasks:

1. Scans all applicable source code to build a dependency tree
2. Show how objects would be affected as developers write code
3. Generate JSON, or build scripts, to automatically build your application changes
4. Generates reports for branches being worked on so project owners can see their application in real time
5. Migrate your code to git

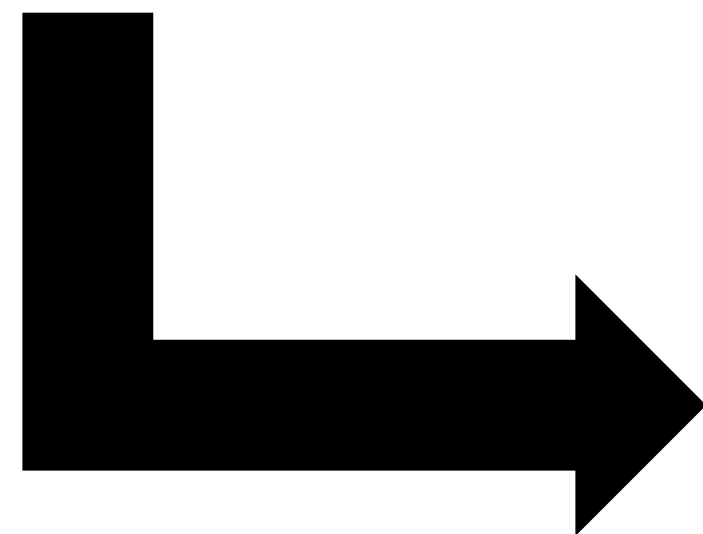
Adds views to show dependency relationships



The screenshot shows the Visual Studio Code interface with the following components:

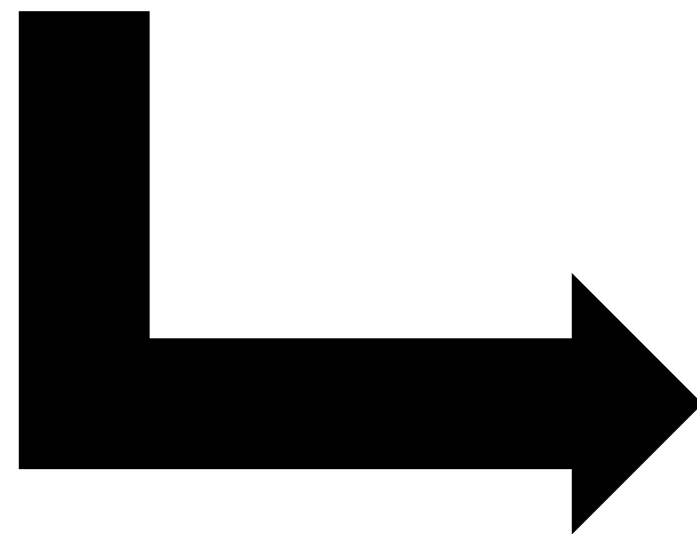
- EXPLORER:** Shows the project structure for 'ibmi-company_system'. The 'qddsrc' folder is expanded, showing files like 'department.table', 'depts.dspf', 'employee.table', 'emps.dspf', 'nemp.dspf', 'popdept.sql', and 'popemp.sql'. The 'employee.table' file is selected.
- OPEN EDITORS:** Shows 'employee.table qddsrc' as the active editor.
- SOURCE IMPACTS:** A panel at the bottom shows the impact of changes to 'employee.table'. It lists:
 - EMPLOYEE.FILE qddsrc\employee.table
 - EMPLOYEES.PGM qrpglesrc\employees.pgm.sqlrpgle
 - DEPTS.PGM qrpglesrc\depts.pgm.sqlrpgle
 - NEWEMP.PGM qrpglesrc\newemp.pgm.sqlrpgle
 - DEPTS.PGM qrpglesrc\depts.pgm.sqlrpgle
- EDITOR:** Displays the SQL code for 'employee.table'. The code includes a 'CREATE OR REPLACE TABLE EMPLOYEE' statement with columns: EMPNO (CHAR(6), NOT NULL), FIRSTNME (VARCHAR(12), NOT NULL), MIDINIT (CHAR(1), NOT NULL), LASTNAME (VARCHAR(15), NOT NULL), WORKDEPT (CHAR(3)), PHONENO (CHAR(4)), HIREDATE (DATE), JOB (CHAR(8)), EDLEVEL (SMALLINT, NOT NULL), SEX (CHAR(1)), BIRTHDATE (DATE), SALARY (DECIMAL(9,2)), BONUS (DECIMAL(9,2)), and COMM (DECIMAL(9,2)). It also includes a 'PRIMARY KEY (EMPNO))' constraint and an 'ALTER TABLE EMPLOYEE' statement to add a foreign key constraint: 'ADD FOREIGN KEY RED (WORKDEPT) REFERENCES DEPARTMENT ON DELETE SET NULL;'. The status bar at the bottom indicates 'Ln 6, Col 45 Spaces: 6 UTF-8 CRLF { } sql'.

Show which files are impacted by the edited file



i.e. changes in the editor could impact these programs which use the table

Show which objects are impacted by changes in the current commit or change



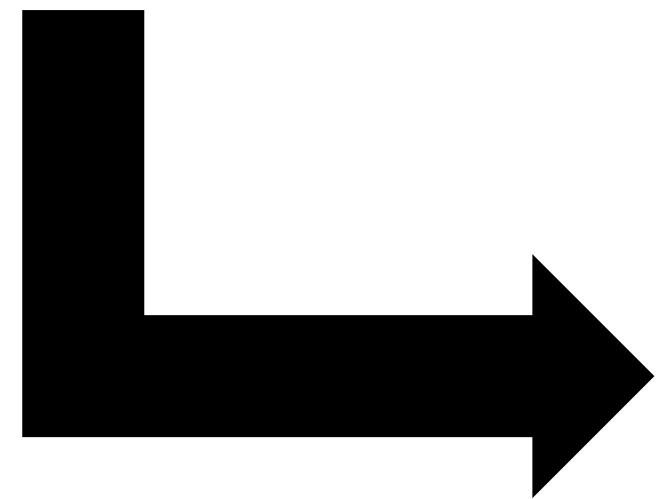
The screenshot shows the Visual Studio Code interface with the following components:

- Source Control Panel (Left):**
 - Repository: `ibmi-company_system` (Git)
 - Commit Message: `Message (Ctrl+Enter to commit on "main")`
 - Commit Button: `Commit`
 - Changes (4):
 - `department.table` (qddssrc) - Modified (M)
 - `nemp.dspf` (qddssrc) - Modified (M)
 - `employees.pgm.sqlrpgle` (qrpglesrc) - Modified (2, M)
 - `mypgm.pgm.rpgle` (qrpglesrc) - Modified (M)
 - Change Impacts:
 - `DEPARTMENT.FILE` (qddssrc\department.table)
 - `POPDEPT.PGM` (qddssrc\popdept.sql)
 - `DEPTS.PGM` (qrpglesrc\depts.pgm.sqlrpgle)
 - `NEMP.FILE` (qddssrc\nemp.dspf)
 - `NEWEMP.PGM` (qrpglesrc\newemp.pgm.sqlrpgle)
 - `DEPTS.PGM` (qrpglesrc\depts.pgm.sqlrpgle)
 - `EMPLOYEES.PGM` (qrpglesrc\employees.pgm.sqlrpgle)
 - `DEPTS.PGM` (qrpglesrc\depts.pgm.sqlrpgle)
 - `MYPGM.PGM` (qrpglesrc\mypgm.pgm.rpgle)
- Code Editor (Right):**
 - File: `mypgm.pgm.rpgle` (Working Tree)
 - Content:

```
1 1 **free
2 2
3 3 ctl-opt dftactgrp(*no);
4 4
5 5 /INCLUDE 'qrppleref/constants.rpgleinc'
6 6
7 7 dcl-s mytext char(50);
8 8
9 9 Dcl-PR printf Int(10) extproc('printf');
10 10 | input Pointer value options(*string);
11 11 End-PR;
12 12
13 - mytext = 'Hello to all you people';
13+ mytext = 'Hello everyone!';
14 14 printf(mytext);
15 15
16 16 dsply mytext;
17 17
18 18 return;
```
- Status Bar (Bottom):**
 - Repository: `main*`
 - Project: `ibmi-company_system`
 - Language: `RPGLE`
 - Cursor: `Ln 18, Col 8`

Integration with IBM i Project Explorer

What is the dependency tree for the given project



Action to fix include syntax and file extension

Action to generate or refresh dependency information as json, make, bob, or mark down

The screenshot displays the IBM i Project Explorer interface within Visual Studio Code. The Project Explorer on the left shows a project structure with a pink box highlighting the 'Source Orbit' section. A tooltip 'Autofix' is visible over the 'Autofix' icon. The main editor shows a COBOL program 'employees.pgm.sqlrpgle' with code including an include statement. The status bar at the bottom shows 'Project: ibmi-company_system'.

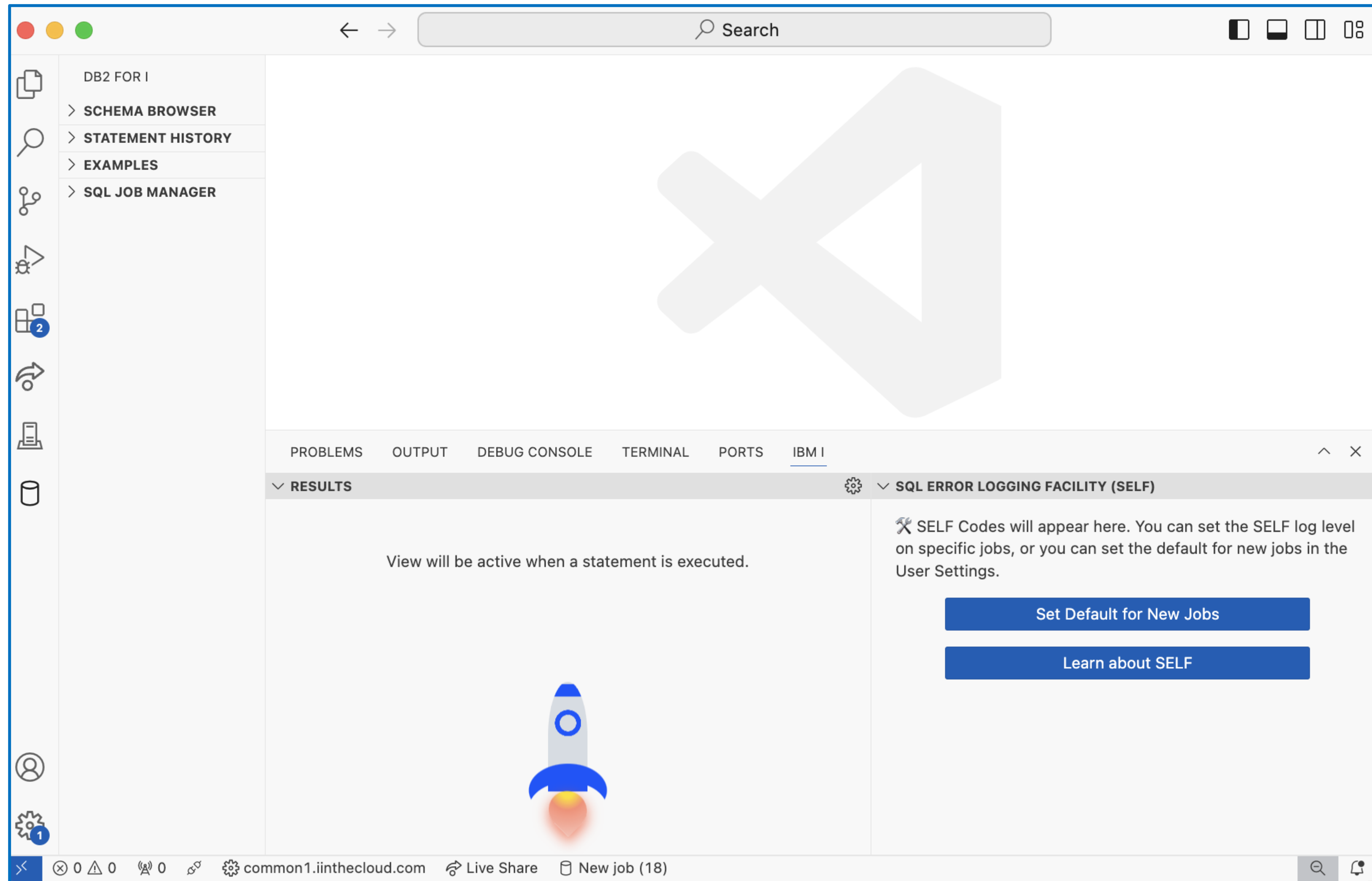
```
qrpglesrc > employees.pgm.sqlrpgle > ClearSubfile
1  **free
2  Ctl-Opt DFTACTGRP(*no);
3
4  Dcl-Pi EMPLOYEES;
5  DEPTNO Char(3);
6  End-Pi;
7
8  | | //-----*
9
10 /include 'qrpgleref/constants.rpgleinc'
11
12 | | //-----*
13
14 Dcl-F emps WORKSTN Sfile(SFLDta:Rrn) IndDS(WkStnInd) InfDS(fileinfo);
15
16 Dcl-S Exit Ind Inz(*Off);
17
18 Dcl-S Rrn          Zoned(4:0) Inz;
19
20 Dcl-DS WkStnInd;
21 ProcessSCF      Ind      Pos(21);
22 ReprintScf      Ind      Pos(22);
23 Error           Ind      Pos(25);
24 PageDown       Ind      Pos(30);
25 PageUp         Ind      Pos(31);
26 SflEnd         Ind      Pos(40);
27 SflBegin       Ind      Pos(41);
28 NoRecord       Ind      Pos(60);
29 SflDspCtl      Ind      Pos(85);
30 SflClr         Ind      Pos(75);
31 SflDsp         Ind      Pos(95);
32 End-DS;
33
34 Dcl-DS FILEINFO;
```


Db2 extension

Powered by Mapepire



Meet the Database Extension



SQL Query and results

```
SELECT * FROM TOYSTORE3.CUST_DIM  
where country = 'CANADA' and salesperson = 'SalesPerson#00007'  
limit 10;
```

RESULTS							
CUSTKEY	CUSTOMER	ADDRESS	PHONE	MKTSEGMENT	COUNTRY	CONTINENT	
13	Customer#000000013	wM1A1C75Qi6Ljx2N6C	13-761-547-5974	BUILDING	CANADA	AMERICA	
22	Customer#000000022	6wn67O2wBh4Qi26MLRSMAOQ2	13-806-545-9701	MACHINERY	CANADA	AMERICA	
226	Customer#000000226	2 2Qxi0B16202gA1C5yyjOSjOOy	13-452-318-7709	AUTOMOBILE	CANADA	AMERICA	
249	Customer#000000249	1hB6hQ2mz00 ICiLgOLg6M770m01y6L	13-226-455-7727	MACHINERY	CANADA	AMERICA	
440	Customer#000000440	C2S7mi25P2hP2CC	13-244-480-5751	MACHINERY	CANADA	AMERICA	
702	Customer#000000702	3OMP4Skw761Onj21i7CAC3 xS5yALCSAn0	13-549-296-5659	FURNITURE	CANADA	AMERICA	

Data Load SQL Generator

DB2 FOR I

and scheduled job info.sql drive kraken77 table dml on ctcdb31a.sql iSee Dead Jobs.sql

1 SELECT * FROM TOYSTORE3.CUST_DIM
2 where country = 'CANADA' and salesperson = 'SalesPerson#00007'
3 limit 10;
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS IBM I

▼ RESULTS

CUSTKEY	CUSTOMER	ADDRESS	PHONE	MI
13	Customer#000000013	wM1A1C75Qi6Ljx2N6C	13-761-547-5974	BL
22	Customer#000000022	6wn67O2wBh4Qi26MLRSMAOQ2	13-806-545-9701	M/
226	Customer#000000226	2 2Qxi0B16202gA1C5yyjOSjOOy	13-452-318-7709	AL
249	Customer#000000249	1hB6hQ2mz00 ICiLgOLg6M770m01y6L	13-226-455-7727	M/
440	Customer#000000440	C2S7mi25P2hP2CC	13-244-480-5751	M/
702	Customer#000000702	3OMP4Skw761Onj21i7CAC3 xS5yALCSAn0	13-549-296-5659	FL
			13-264-	

> STATEMENT HISTORY

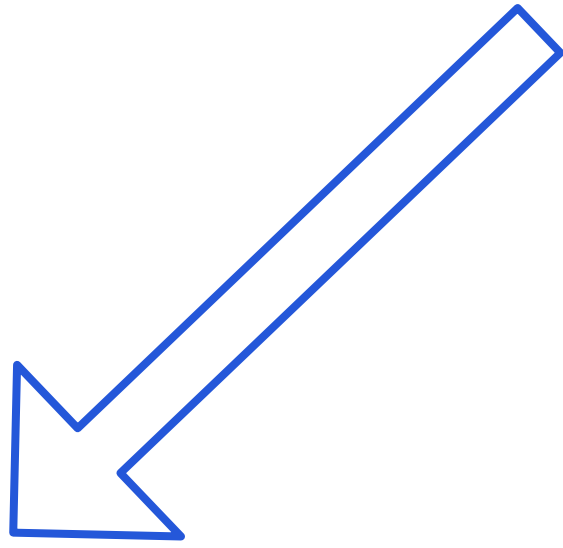
> EXAMPLES

▼ SQL JOB ...

- New job (14) 262454/...
- ▼ Saved Configuration
 - + ctcdb31a - db2data...
 - + common1 SQL nami...
 - + sync19 - iasp1

**Generate an INSERT with VALUES
for this data**

Prefix the query with sql:



```
sql: SELECT * FROM TOYSTORE3.CUST_DIM  
where country = 'CANADA' and salesperson = 'SalesPerson#00007'  
limit 10;
```

Code4i built the INSERT with VALUES SQL statement

-- Generated 1 insert statement

```
insert into TABLE (  
  CUSTKEY, CUSTOMER, ADDRESS, PHONE, MKTSEGMENT, COUNTRY, CONTINENT, REGION, TERRITORY, SALESPERSON, DUMMYKEY  
) values  
(13, 'Customer#00000013', 'wM1A1C75Qi6Ljx2N6C', '13-761-547-5974', 'BUILDING', 'CANADA', 'AMERICA',  
'NORTHEAST', 'THREE', 'SalesPerson#00007', '*'),  
(22, 'Customer#00000022', '6wn6702wBh4Qi26MLRSMA0Q2', '13-806-545-9701', 'MACHINERY', 'CANADA',  
'AMERICA', 'NORTHWEST', 'ONE', 'SalesPerson#00007', '*'),  
(226, 'Customer#000000226', '2 2Qxi0B16202gA1C5yyj0Sj00y', '13-452-318-7709', 'AUTOMOBILE', 'CANADA',  
'AMERICA', 'SOUTHEAST', 'FIVE', 'SalesPerson#00007', '*'),  
(249, 'Customer#000000249', '1hB6hQ2mz00 lCiLg0Lg6M770m01y6L', '13-226-455-7727', 'MACHINERY', 'CANADA',  
'AMERICA', 'SOUTHWEST', 'FOUR', 'SalesPerson#00007', '*'),  
(440, 'Customer#000000440', 'C2S7mi25P2hP2CC', '13-244-480-5751', 'MACHINERY', 'CANADA', 'AMERICA',  
'NORTHWEST', 'FIVE', 'SalesPerson#00007', '*'),
```

Data Load SQL Generators

```
sql: SELECT *  
FROM SAMPLE.EMPLOYEE  
WHERE EMPNO != '000010'
```

```
csv: SELECT *  
FROM SAMPLE.EMPLOYEE  
WHERE EMPNO != '000010'
```

```
json: SELECT *  
FROM SAMPLE.EMPLOYEE  
WHERE EMPNO != '000010'
```

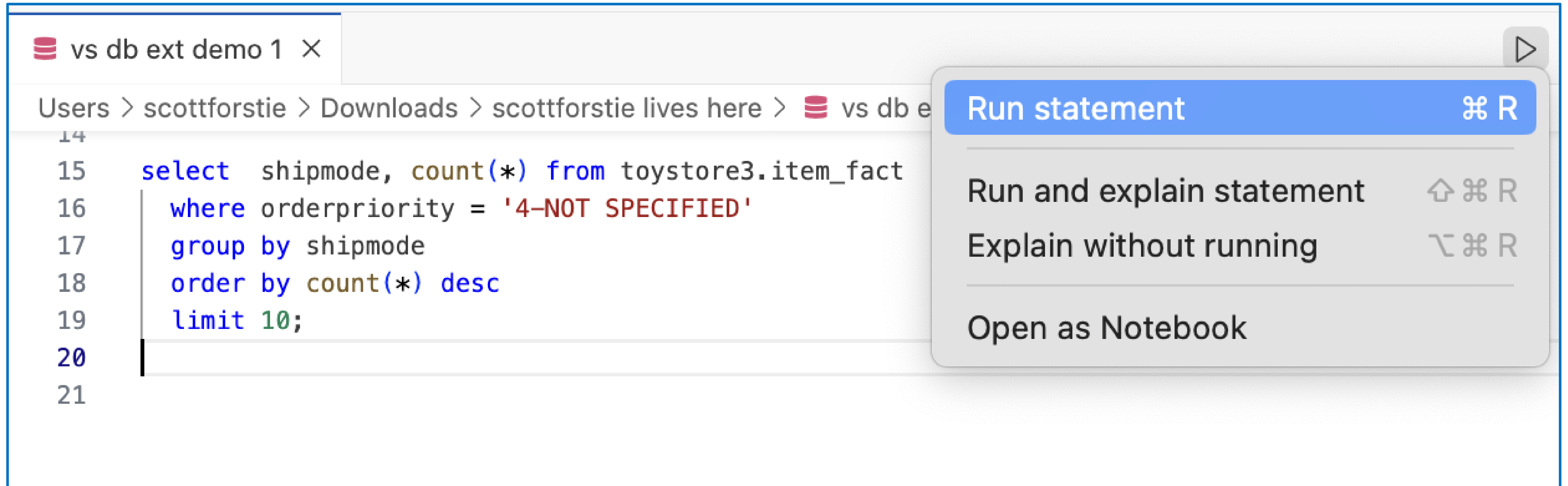
SQL Query and results published in JSON

```
json: SELECT * FROM TOYSTORE3.CUST_DIM  
where country = 'CANADA' and salesperson = 'SalesPerson#00007'  
limit 10;
```

```
{  
  "CUSTKEY": 13,  
  "CUSTOMER": "Customer#000000013",  
  "ADDRESS": "wM1A1C75Qi6Ljx2N6C",  
  "PHONE": "13-761-547-5974",  
  "MKTSEGMENT": "BUILDING",  
  "COUNTRY": "CANADA",  
  "CONTINENT": "AMERICA",  
  "REGION": "NORTHEAST",  
  "TERRITORY": "THREE",  
  "SALESPERSON": "SalesPerson#00007",  
  "DUMMYKEY": "*"  
},  
{  
  "CUSTKEY": 22,  
  "CUSTOMER": "Customer#000000022",
```


SQL Execution Options – Run

Execute the statement



The screenshot shows a SQL editor window titled "vs db ext demo 1". The breadcrumb path is "Users > scottforstie > Downloads > scottforstie lives here > vs db e". The SQL query is:

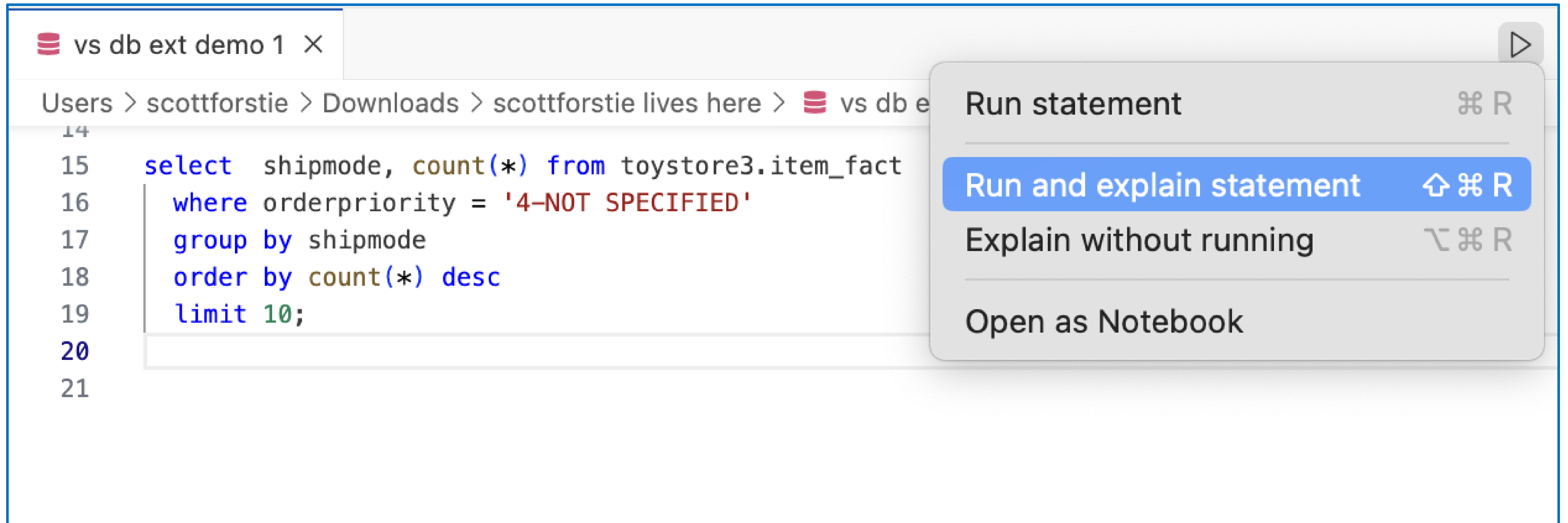
```
14  
15 select shipmode, count(*) from toystore3.item_fact  
16 where orderpriority = '4-NOT SPECIFIED'  
17 group by shipmode  
18 order by count(*) desc  
19 limit 10;  
20  
21
```

A context menu is open over the query, listing the following options:

- Run statement ⌘ R
- Run and explain statement ⌘ R
- Explain without running ⌘ R
- Open as Notebook

SQL Execution Options – Run and Explain

Execute and use Visual Explain to understand the query implementation



The screenshot shows a SQL editor window titled "vs db ext demo 1". The breadcrumb navigation is "Users > scottforstie > Downloads > scottforstie lives here > vs db e". The SQL query is as follows:

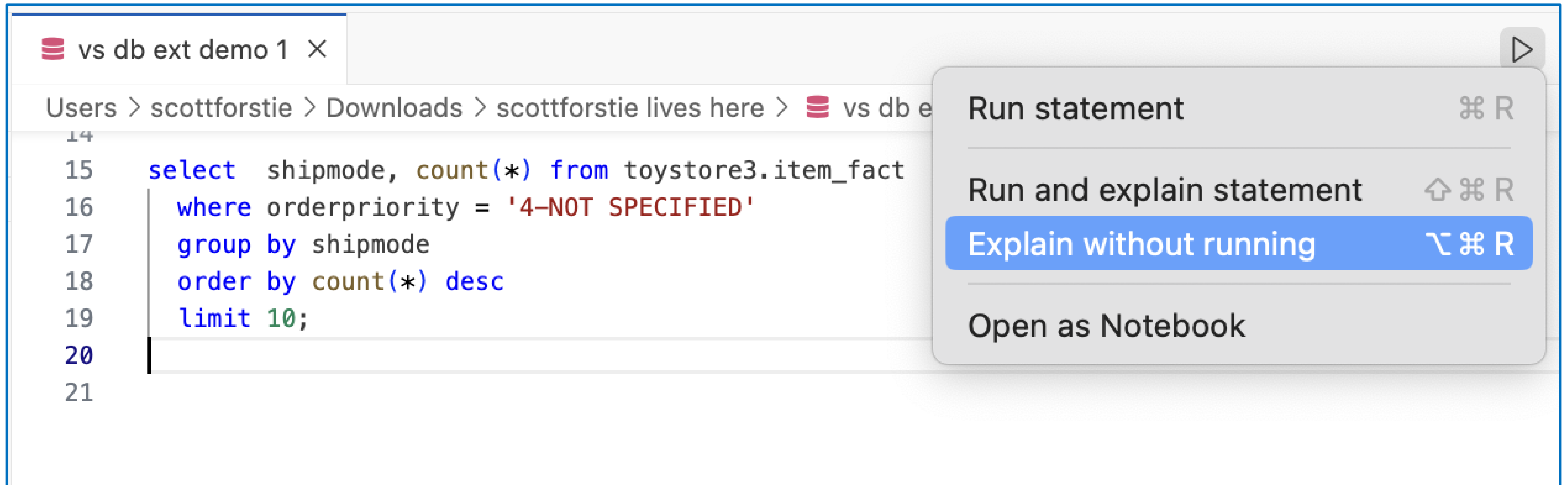
```
14  
15 select shipmode, count(*) from toystore3.item_fact  
16 where orderpriority = '4-NOT SPECIFIED'  
17 group by shipmode  
18 order by count(*) desc  
19 limit 10;  
20  
21
```

A context menu is open over the query, listing the following options:

- Run statement ⌘ R
- Run and explain statement** ⌘ R (highlighted)
- Explain without running ⌘ R
- Open as Notebook

SQL Execution Options – Explain

Use Visual Explain to understand the possible query implementation



The screenshot shows a SQL editor window titled "vs db ext demo 1". The breadcrumb path is "Users > scottforstie > Downloads > scottforstie lives here > vs db e". The SQL query is:

```
14  
15 select shipmode, count(*) from toystore3.item_fact  
16 where orderpriority = '4-NOT SPECIFIED'  
17 group by shipmode  
18 order by count(*) desc  
19 limit 10;  
20  
21
```

A context menu is open over the query, listing the following options:

- Run statement ⌘ R
- Run and explain statement ⌘ R
- Explain without running ⌘ R** (highlighted in blue)
- Open as Notebook

Db2 Extension within Code for i

The screenshot displays the IBM i development environment. The top pane shows a SQL query:

```
1 SELECT USER_NAME, SUM(SIZE) AS TOTAL_SPOOL_SPACE FROM  
2 ... TABLE (QSYS2.OBJECT_STATISTICS('QSYS...', '*LIB')) AS a,  
3 ... TABLE (QSYS2.OBJECT_STATISTICS(a.objname, 'OUTQ')) AS b,  
4 ... TABLE (QSYS2.OUTPUT_QUEUE_ENTRIES(a.objname, b.objname, '*NO')) AS c  
5 WHERE USER_NAME NOT LIKE 'Q%'  
6 GROUP BY USER_NAME  
7 ORDER BY TOTAL_SPOOL_SPACE DESC  
8 LIMIT 10
```

The bottom pane shows the **VISUAL EXPLAIN** tool. The left side lists the execution plan steps:

- Final Select (1 row)
- Fetch N Rows (1 row)
- Sorted List Scan
- Temporary Sorted List
- Hash Scan
- Temporary Distinct Hash Table
- Nested Loop Join (1 row)
- Nested Loop Join
- Cache Probe
- Cache (1 row)
- User Defined Table Function qsys2.object_s...
- Cache Probe
- Cache
- User Defined Table Function qsys2.object_s...
- Cache Probe
- Cache
- User Defined Table Function qsys2.output_q...

The right side of the Visual Explain tool shows details for the selected step:

- Estimated Time Information (Start Time, Total Time)**: Cumulative Time(ms) 7.036
- Cache Information**: Estimated Total Cache Size(Bytes) 139,264; Rows in Cache 5,000; Number of Cache Points 1; Key Cardinality 1; Key Size(bytes) 0; Non Key Data Size(bytes) 19; Populate At Open No; Temporary Reusable No
- Estimated Cost Information About the Plan Performed**: Estimated Population Time 1.117; I/O Or CPU Bound CPU Bound; CPU Cost(ms) 1.117

A blue callout box with the text "Visual Explain Built-in!" has an arrow pointing to the Visual Explain tool. Another blue arrow points from the SQL editor to the "Run statement" button in the top right.

Insert from Examples

EXAMPLES

- Built-in Global Variables
- Data Control Language (DCL)
- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Db2 for i Services
- Geospatial Analytics
 - ST_Area
 - ST_AsBinary
 - ST_AsText
 - ST_Buffer
 - ST_Contains
 - ST_Covers
 - ST_Crosses
 - ST_Difference
 - ST_Disjoint

SQL JOB MANAGER

```
3 -- Create a table with a polygon column and insert several po
4 CREATE TABLE sample_parks (park_name VARCHAR(50), geometry QS
5 INSERT INTO sample_parks (park_name, geometry) VALUES
6   ('Central Park', QSYS2.ST_POLYGON('polygon((-73.9817 40.768
7   ('Washington Square Park', QSYS2.ST_POLYGON('polygon((-73.9
8
9 -- Find the area of each New York City Park in square meters
10 SELECT park_name,
11         QSYS2.ST_AREA(geometry) as area_square_meters,
12         QSYS2.ST_AREA(geometry) * 0.000247 as area_acres
13 FROM sample_parks;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS IBM I

RESULTS

PARK_NAME	AREA_SQUARE_METERS	AREA_ACRES
Central Park	3427480.9038014133	846.587783238949
Washington Square Park	40003.49153390578	9.880862408874727

Db2 for IBM i Visual Studio Code Extension



Db2 for IBM i v1.5.1

Code4i | 20,678 | ★★★★★ (4)

Db2 for IBM i tools in VS Code

Disable **Uninstall** Auto Update ⚙️

Code for IBM i: Connect to IBM i and write RPG, CL, and more from VS Code

Db2 for IBM i: Connect to IBM i and write SQL

- Run SQL statements
- View database objects (schema browser)
- Visual Explain
- **SQL Error Logging Facility**
- Notebooks

**Start using the
Code4i
Database Extension**

<https://marketplace.visualstudio.com/items?itemName=HalcyonTechLtd.vscode-db2i>

VsCode Database Extension & SELF

- Opt in or out of SELF

The screenshot shows the VS Code interface with the 'vscode-db2i' extension installed. The left sidebar displays the 'SCHEMA BROWSER' and 'STATEMENT HISTORY' panels. The main editor area shows the 'User' settings for the extension, with a blue box highlighting the 'vscode-db2i' extension name. A second blue box highlights the 'Vscode-db2i: Job Self Default' configuration, which is set to '*ALL'.

DB2 FOR I

SCHEMA BROWSER

- > "lower"
- > "thelib"
- > QSYS
- > QSYS2
- > SYSTOOLS
- > THE_TOP_OF_THINGS_YA
- > TOYSTORE3

STATEMENT HISTORY

- > Past day
 - call coolstuff.proc1()
 - values sysibmadm.selfc...
 - call coolstuff.proc6()
 - call coolstuff.proc5()
 - call coolstuff.proc3()
 - call coolstuff.proc4()

selfie1.sql

Untitled-2

iSee SELF enhancements (SQL Error Logging Facility) in the VS Co

vscode-db2i

User

- Text Editor (2)
- Features (1)
 - Notebook (1)
- Extensions (10)
 - Db2 for IBM i (10)
 - General (3)
 - SQL Formatting (2)
 - Viewing Data (2)
 - Visual Explain (1)
 - SQL Error Logging Facility (SELF) (2)

Vscode-db2i: Job Self Default

Default SELF setting for new jobs

- *ALL
- *NONE default
- *ALL
- *ERROR
- *WARNING

<https://www.ibm.com/support/pages/node/6828091>

VsCode Database Extension & SELF

- Enable or Disable auto-refresh of SELF data

The screenshot shows the VS Code interface with the 'Vscode-db2i' settings page open. The left sidebar displays the 'SCHEMA BROWSER' and 'STATEMENT HISTORY' panels. The main editor area shows the 'User' settings for 'vscode-db2i'. A blue box highlights the 'Vscode-db2i: Job Self View Auto Refresh' setting, which is checked and labeled 'Enable auto-refresh for SELF Codes view when connecting to a system'. Other settings visible include 'Vscode-db2i: Job Self Default' set to '*ALL'.

<https://www.ibm.com/support/pages/node/6828091>

VsCode Database Extension & SELF

Challenge: Finding, understanding, and fixing SQL errors and warnings

SELF - Captures details about user specified SQL errors or warnings

VsCode Db2 for i extension provides a new easy to use interface for seeing the item captured by SELF

The screenshot displays the VS Code interface with the Db2 for i extension. The main editor shows SQL code with comments like "-- Cause an SQL warning" and "-- Force SQL warnings". The left sidebar shows the Schema Browser and Statement History. The bottom panel shows the RESULTS window with the SQL ERROR LOGGING FACILITY (SELF) results.

SQL Code:

```
30  
31 --  
32 -- Cause an SQL warning  
33 --  
34 create or replace procedure coolstuff.proc1()  
35 begin  
36 declare v1 integer;  
37 declare v2 integer;  
38  
39 select 1+1, 2+2, 3+3 into v1, v2  
40 | from sysibm.sysdummy1;  
41  
42 end;  
43 stop;  
44  
45 --  
46 -- Force SQL warnings  
47 --  
48 call coolstuff.proc1();  
49 stop;  
50 |  
51 --  
52 -- Cause an SQL warning  
53 --  
54 create or replace procedure coolstuff.proc2()  
55 begin  
56 declare v1 integer;
```

SELF Results:

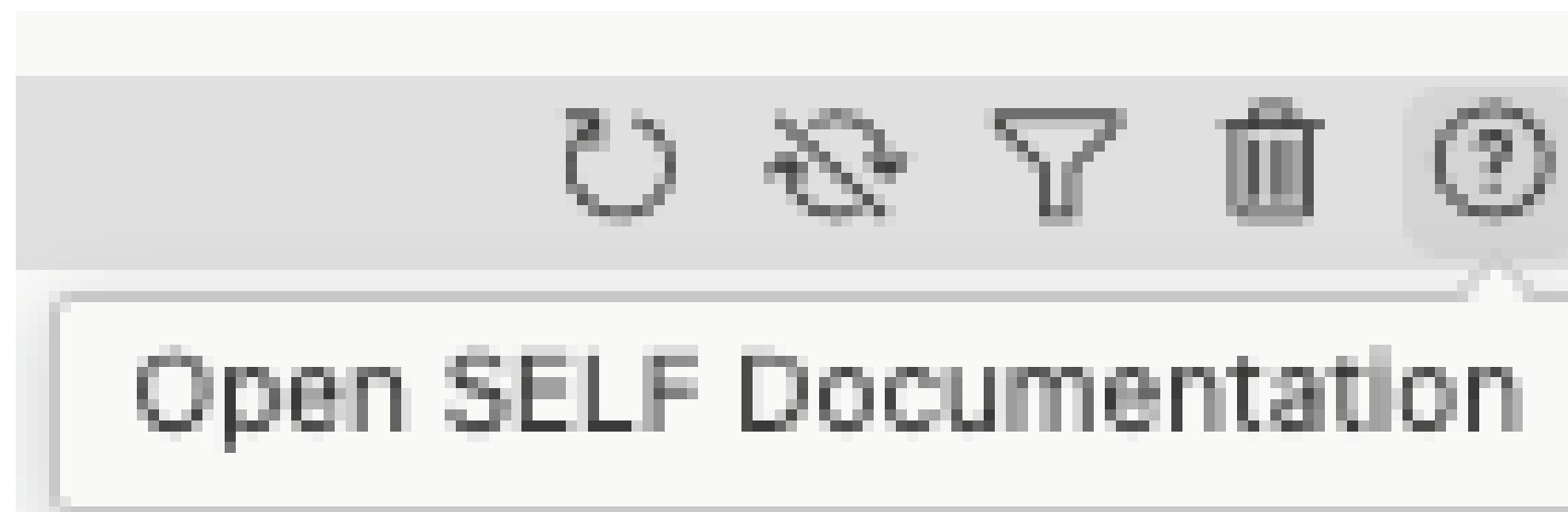
Message ID	Message Text	Count
01503 (30)	Number of host variables less than result values.	5
01503 (30)	Number of host variables less than result values.	5
42908 (-153)	Column list required.	1
42908 (-153)	Column list required.	1
07001 (-313)	Number of host variables not valid.	1
07001 (-313)	Number of host variables not valid.	1
42704 (-204)	&1 in &2 type *&3 not found.	1
42806 (-303)	Variable &2 not compatible or value too long.	1
42806 (-303)	Variable &2 not compatible or value too long.	1
42704 (-204)	&1 in &2 type *&3 not found.	1
01545 (12)	Correlation without qualification occurred for column &1 to table &2.	1

VsCode Database Extension & SELF

- SELF control(s)



- SELF control(s)



All products / IBM i / 7.5 /

K

SQL Error Logging Facility (SELF)

Last Updated: 2024-05-07

The SQL Error Logging Facility (SELF) provides a mechanism that can be used to understand SQL errors or warnings. SELF is built into Db2® for i and can be enabled in specific jobs or

SELF differs from the Db2 for i SQL Performance Monitoring or Database Monitoring features

1. SELF is configured by the client to identify specific SQLCODE values that Db2 for i shows when a SQL statement completes with a matching SQLCODE value.

VsCode Database Extension & SELF

```
93 create or replace procedure coolstuff.proc4()
94 begin
95 declare v1 integer;
96 declare v2 integer;
97
98 declare demoC cursor for
99 select 1, current_date
100 | from sysibm.sysdummy1;
101 open demoC;
102 fetch from demoC into v1, v2;
103 close demoC;
104 end;
105 stop;
106
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS IBM I

RESULTS

Statement executed with no result set returned. Rows affected: 0

SQL ERROR LOGGING FACILITY (SELF)

- > ⚠ 01503 (30) Number of host variables less than result values.
- > ⚠ 01503 (30) Number of host variables less than result values.
- > ⊗ 42908 (-153) Column list required.
- > ⊗ 42908 (-153) Column list required.
- > ⊗ 07001 (-313) Number of host variables not valid.
- > ⊗ 07001 (-313) Number of host variables not valid.
- > ⊗ 42704 (-204) &1 in &2 type *&3 not found.
- > ⊗ 42806 (-303) Variable &2 not compatible or value too long.
- ∨ ⊗ 42806 (-303) Variable &2 not compatible or value too long.
 - 📄 Statement FETCH FROM DEMOC INTO : H : H , : H : H
 - 🕒 Job 248596/QUSER/QZDASOINIT
 - 🏠 Client Name vscode-db2i 1.5.1
 - 🏠 Client Program VSCode connector | Version 2.1.4
 - 🏠 Object COOLSTUFF/PROC4 (*PGM, PROC4)
- ∨ 📄 Stack
 - _C_peg:undefined COOLSTUFF/PROC4 (ILE, PROC4)
 - main:0000000033 COOLSTUFF/PROC4 (ILE, PROC4)

Notebooks

SQL Visualizations in Graph form!

The screenshot shows the IBM i Notebook interface. On the left is a sidebar with a 'SCHEMA BROWSER' and 'STATEMENT HISTORY' sections. The 'STATEMENT HISTORY' section shows a list of SQL statements, with the first one selected: 'SELECT USER_NAME, SUM(SIZE) A...'. Below this is an 'EXAMPLES' section with various categories like 'Built-in Global Variables', 'Data Control Language (DCL)', etc. The 'Notebooks' category is highlighted with a red arrow.

The main area displays a code editor with a SQL query:

```
line: SELECT USER_NAME as label, SUM(SIZE) AS TOTAL_SPOOL_SPACE  
TABLE (QSYS2.OBJECT_STATISTICS('QSYS', '*LIB') ) as a,  
TABLE (QSYS2.OBJECT_STATISTICS(a.objname, 'OUTQ') ) AS b,  
TABLE (QSYS2.OUTPUT_QUEUE_ENTRIES(a.objname, b.objname, '*NO')) AS c  
WHERE USER_NAME NOT LIKE 'Q%'  
GROUP BY USER_NAME  
ORDER BY TOTAL_SPOOL_SPACE DESC  
FETCH FIRST 10 ROWS ONLY
```

Below the code editor, the query is executed, and the results are visualized as a line graph. The graph shows 'TOTAL_SPOOL_SPACE' on the y-axis (ranging from 0 to 300,000) and user names on the x-axis. The data points are: MCCARGAR (~280,000), DAWNMM (~220,000), SCOTTFF (0), TIMMR (0), WLAB01 (0), WLAB06 (0), WLAB05 (0), JULIAYAN (0), REINHARD (0), and AECIESLA (0). A pink box labeled 'SQL' points to the query text, and another pink box labeled 'Graph' points to the visualization.

User Name	TOTAL_SPOOL_SPACE
MCCARGAR	~280,000
DAWNMM	~220,000
SCOTTFF	0
TIMMR	0
WLAB01	0
WLAB06	0
WLAB05	0
JULIAYAN	0
REINHARD	0
AECIESLA	0

```
set LOCAL_SQLCODE = 0;
while (lc < loop_var) do
  call qsys2.ifs_write(
    path_name
    line =>
    overwrite
    end_of_line
  );
  set lc = lc + 1;
end while;
end;
```

```
create or replace
  modifies sql
  set option c
begin
  declare LOCAL_
  declare LOCAL_
  declare V_Mess
```

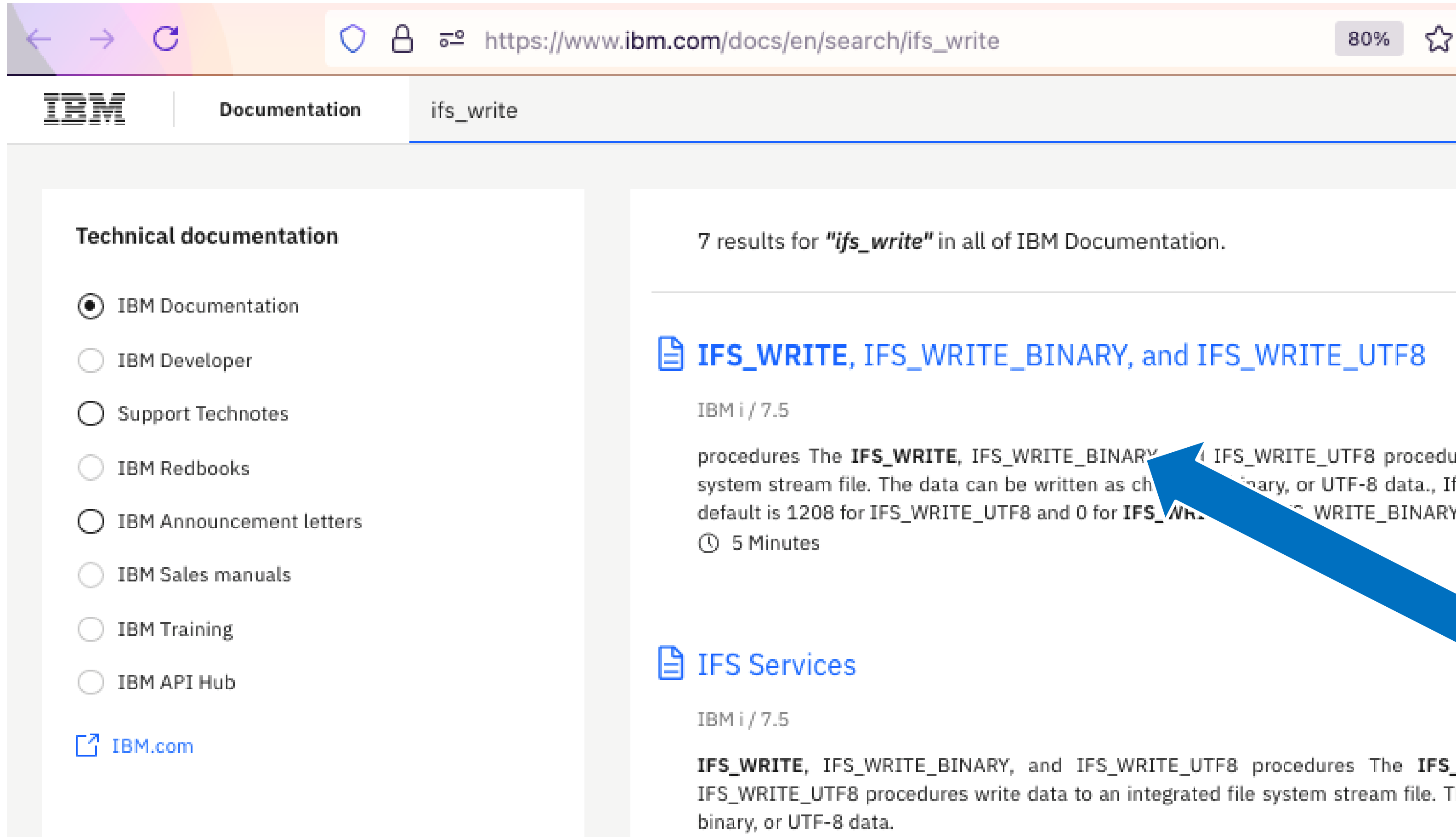
ifs_write(
 path_name => double-byte character large object(16777216),
 line => character large object(2147483647),
 file_ccsid => integer(10, 0),
 overwrite => character varying(7),
 end_of_line => character varying(4)
)

@param path_name pathname - Default: none
@param line string - Default: none
@param file_ccsid number - Default: 0
@param overwrite APPEND, NONE, REPLACE - Default: APPEND
@param end_of_line ANY, CR, CRLF, LF, LFCR, NONE - Default: CRLF

[Search on IBM Documentation](#)

Hover Support:

- QSYS, QSYS2, and SYSTOOLS schemas
- Functions, Procedures, and Views
- Parameter detail
- [Link to IBM i Documentation](#)



The screenshot shows a web browser window with the URL https://www.ibm.com/docs/en/search/ifs_write. The page title is "ifs_write" under the "Documentation" tab. On the left, there is a sidebar with "Technical documentation" and a list of categories: IBM Documentation (selected), IBM Developer, Support Technotes, IBM Redbooks, IBM Announcement letters, IBM Sales manuals, IBM Training, and IBM API Hub. The main content area shows "7 results for 'ifs_write' in all of IBM Documentation." The first result is a document titled "IFS_WRITE, IFS_WRITE_BINARY, and IFS_WRITE_UTF8" from IBM i / 7.5. A blue arrow points to the word "IFS_WRITE" in the document title. Below it is another result titled "IFS Services" from IBM i / 7.5.

Hover Support:

- QSYS, QSYS2, and SYSTOOLS schemas
- Functions, Procedures, and Views
- Parameter detail
- Link to IBM i Documentation

AI Extensions

”AI Won’t Replace Humans — But Humans With AI Will Replace Humans Without AI”

Karim Lakhani, professor at Harvard Business



AI to help people write code?

Anecdotal observations:

- AI-driven code assistance is already very popular
- Many companies are interested in self-hosting models to avoid sending sensitive information to the cloud

IBM's two-pronged approach:

1. Support community momentum with open solutions and models
 - Synergize with IBM's open source AI strategy
<https://developer.ibm.com/technologies/opensource-ai/>
 - Enable AI for IBM i devs in 2024
2. Work toward a premium IBM offering (WCAi)
 - Leverage large brain trust in IBM i community
 - Enable more advanced modernization activities

Poll: AI for code assistance for IBM i?

- GitHub CoPilot
- TabNine CoPilot
- Open Source IDE extensions (Codeium, continue, etc)
- ChatGPT or OpenAI web interface
- Other
- None

Known LLM providers

IBM Watsonx.ai

Anthropic

Azure

Mistral

Cohere

Groq

OpenAI

TogetherAI

Google (Gemini)

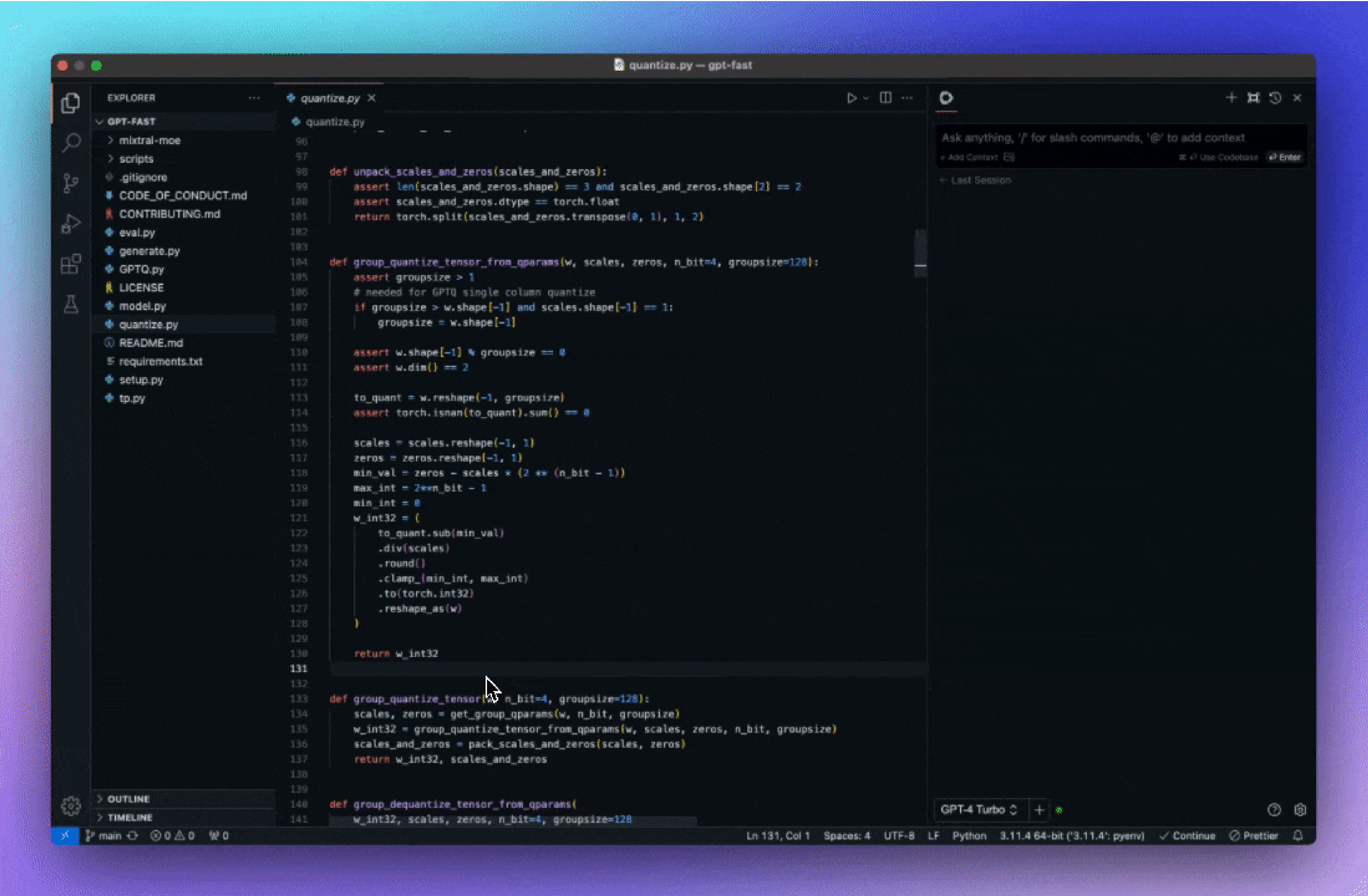
Replicate

..... many more!!

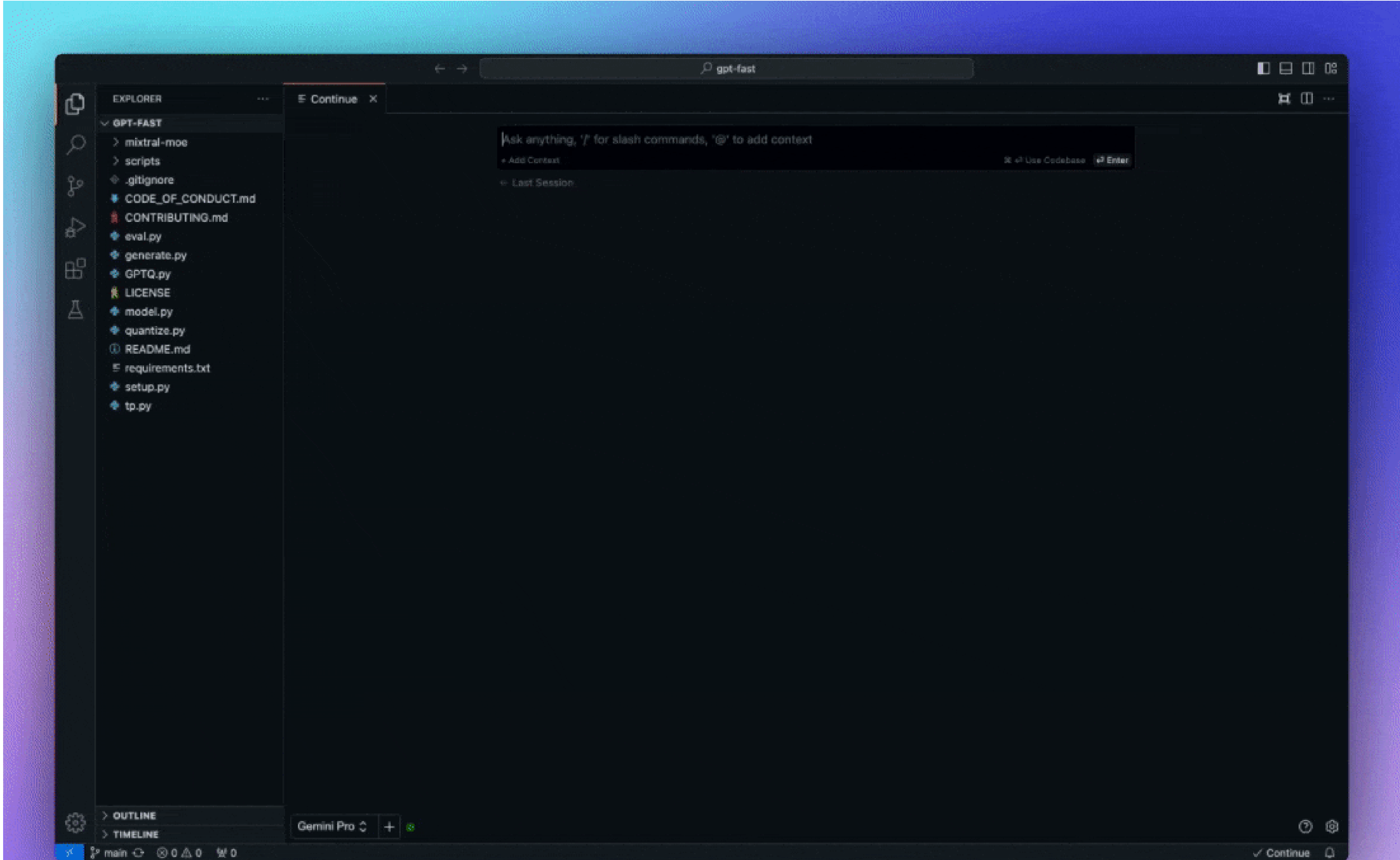


watsonx

Continue: Understand Code sections



Continue: Ask about Codebase



Continue: Chat, Add Files as context

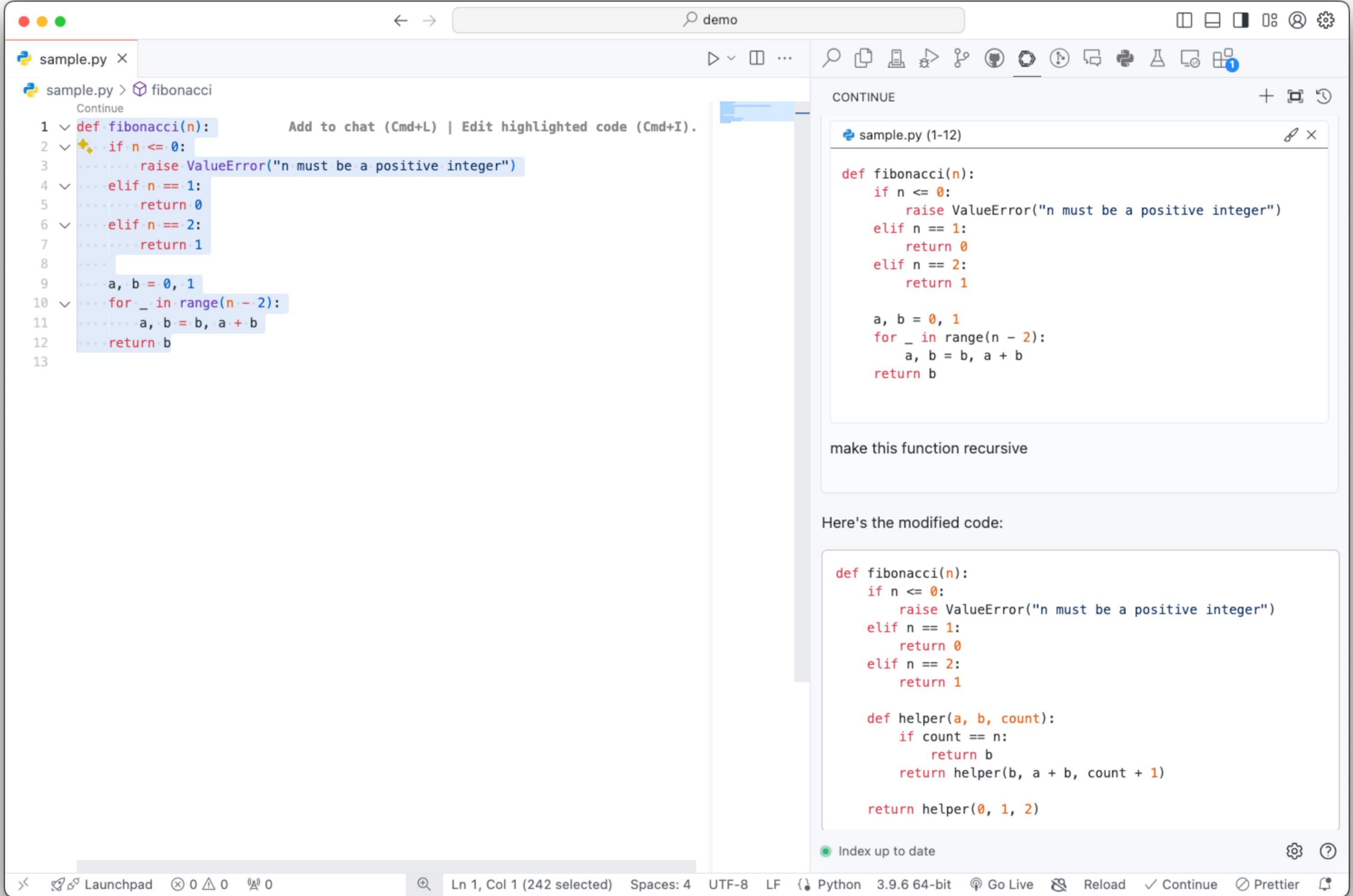
The screenshot shows a code editor with a file named `sample.py` containing a `fibonacci` function. The function is defined as follows:

```
1 def fibonacci(n):
2     if n <= 0:
3         raise ValueError("n must be a positive integer")
4     elif n == 1:
5         return 0
6     elif n == 2:
7         return 1
8
9     a, b = 0, 1
10    for _ in range(n - 2):
11        a, b = b, a + b
12    return b
13
```

The chat interface on the right shows the prompt: `@sample.py explain how this code works`. The response provides a detailed explanation of the function's logic, including the initial conditions, the `if-elif-else` block, the `for` loop, and the final return statement.

“@sample.py”
adds sample.py to
prompt

Continue: add selected code to context



The screenshot shows a code editor with a Python file named 'sample.py'. The code defines a 'fibonacci' function with a loop. A 'Continue' chat interface is open on the right, showing a conversation about making the function recursive. The chat includes a prompt, a response with modified code, and a confirmation of the changes.

```
def fibonacci(n):  
    if n <= 0:  
        raise ValueError("n must be a positive integer")  
    elif n == 1:  
        return 0  
    elif n == 2:  
        return 1  
    a, b = 0, 1  
    for _ in range(n - 2):  
        a, b = b, a + b  
    return b
```

CONTINUE

sample.py (1-12)

```
def fibonacci(n):  
    if n <= 0:  
        raise ValueError("n must be a positive integer")  
    elif n == 1:  
        return 0  
    elif n == 2:  
        return 1  
  
    a, b = 0, 1  
    for _ in range(n - 2):  
        a, b = b, a + b  
    return b
```

make this function recursive


Here's the modified code:

```
def fibonacci(n):  
    if n <= 0:  
        raise ValueError("n must be a positive integer")  
    elif n == 1:  
        return 0  
    elif n == 2:  
        return 1  
  
    def helper(a, b, count):  
        if count == n:  
            return b  
        return helper(b, a + b, count + 1)  
  
    return helper(0, 1, 2)
```

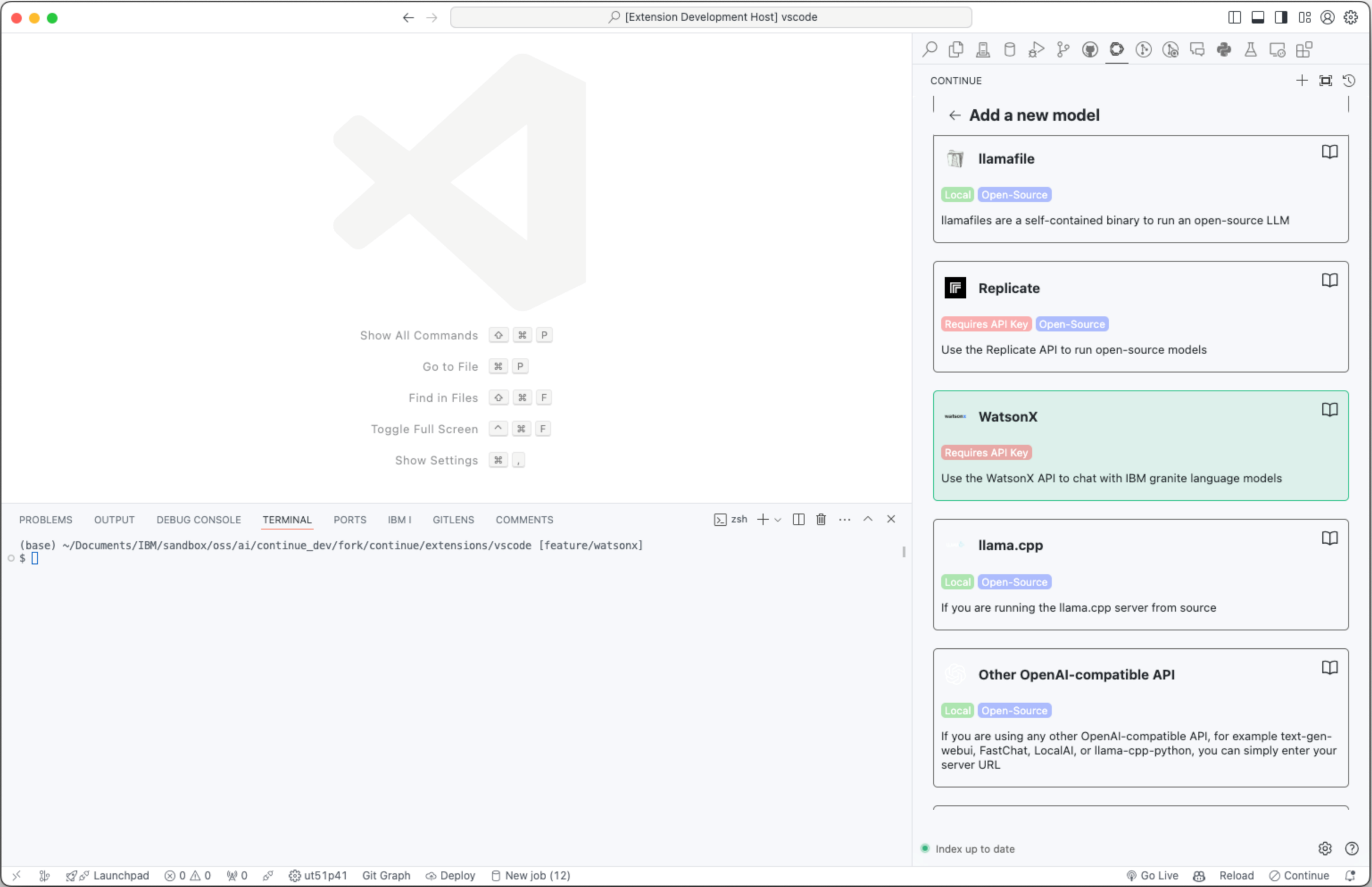
Index up to date

Ln 1, Col 1 (242 selected) Spaces: 4 UTF-8 LF Python 3.9.6 64-bit Go Live Reload ✓ Continue Prettier

IBM enhancements

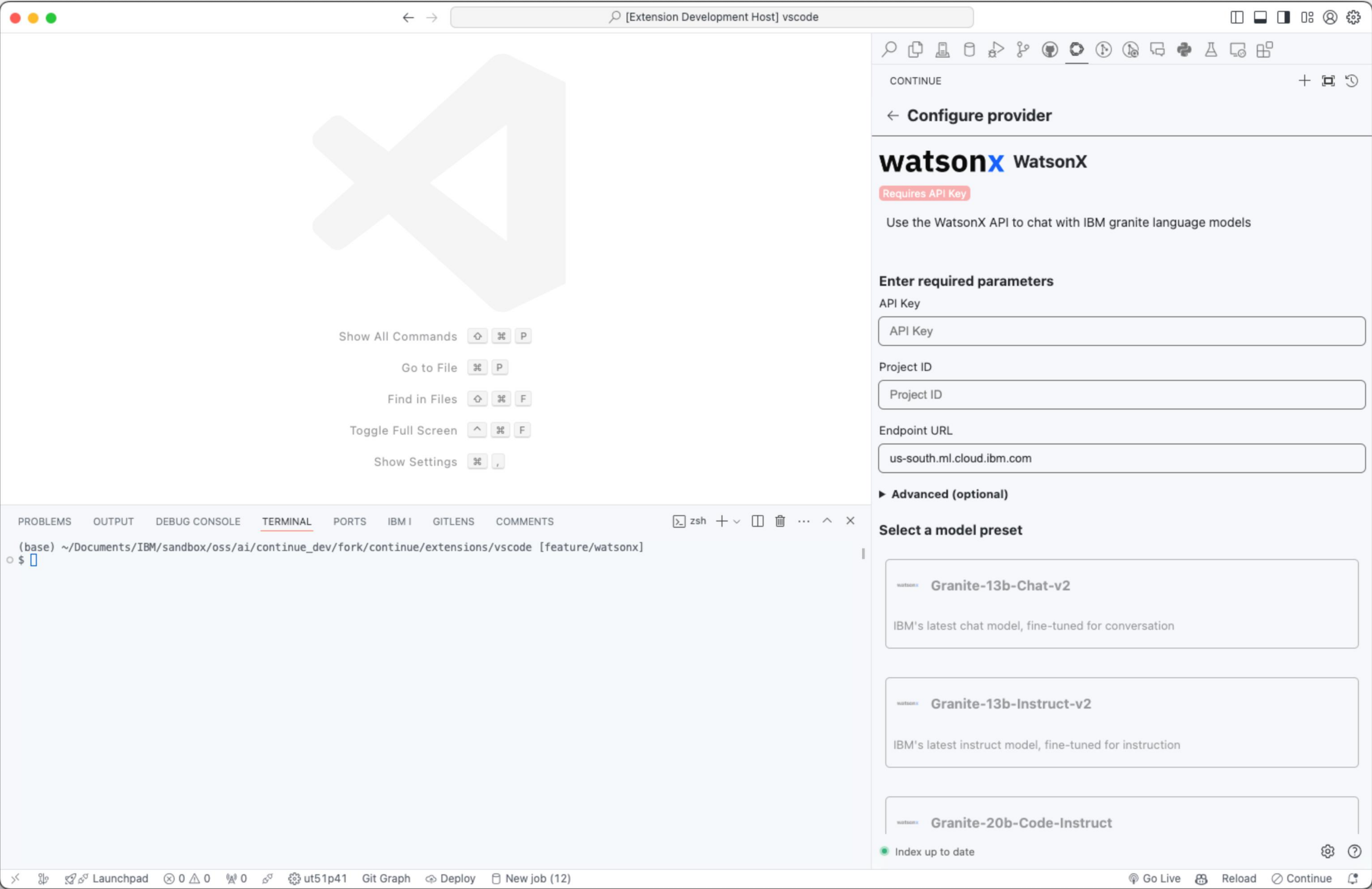
 Continue **+** **watsonx**

Watsonx + Continue



Watsonx.ai model provider in Continue

Watsonx + Continue



The screenshot shows the VS Code interface with the 'Configure provider' dialog for WatsonX. The dialog is titled 'watsonx WatsonX' and includes a red warning 'Requires API Key'. Below this, it says 'Use the WatsonX API to chat with IBM granite language models'. Under 'Enter required parameters', there are three input fields: 'API Key', 'Project ID', and 'Endpoint URL' (pre-filled with 'us-south.ml.cloud.ibm.com'). There is also an 'Advanced (optional)' section. Under 'Select a model preset', there are three options: 'Granite-13b-Chat-v2' (IBM's latest chat model), 'Granite-13b-Instruct-v2' (IBM's latest instruct model), and 'Granite-20b-Code-Instruct'. At the bottom, there is a green dot and the text 'Index up to date'. The background shows the VS Code editor with a large 'X' watermark and a terminal window at the bottom.

Add API key, and Project ID to configure Granite models



- Mixtral 8X7b
- Mixtral 8X7b quantized (deprecated)
- Merlinite (InstructLab Mistral 7b)

NEW Mistral AI Commercial Models

IBM watsonx | Upgrade | MARYAM ASHOORI's Acco... | Dallas | MA

Projects / Sandiego / Prompt Lab | Unsaved | New prompt + | AI guardrails off

Select a foundation model

Select a model that best fits your needs. All models support English text. Check the model information for other supported languages.

Search for a model or task

Filter	Task	Provider	Model type	Model Name	Description	Provider	Type
<input type="checkbox"/> Chat (12)	<input type="checkbox"/> Chat (12) <input type="checkbox"/> Question answering (16) <input type="checkbox"/> Summarization (19) <input type="checkbox"/> Retrieval-Augmented Generation (15) <input type="checkbox"/> Classification (19) <input type="checkbox"/> Generation (19) <input type="checkbox"/> Code generation and conversion (8) <input type="checkbox"/> Extraction (19)	<input type="checkbox"/> BigScience (1)	<input checked="" type="radio"/> All (20)	granite-13b-chat-v2	The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for...	IBM	InstructLab
<input type="checkbox"/> Question answering (16)		<input type="checkbox"/> Code Llama (1)	<input type="radio"/> Provided model (16)	mt0-xxl-13b	An instruction-tuned iteration on mt5.	BigScience	Provided mo...
<input type="checkbox"/> Summarization (19)		<input type="checkbox"/> Google (3)	<input type="radio"/> InstructLab (4)	codellama-34b-instruct-hf	Code Llama is an AI model built on top of Llama 2, fine-tuned for generating and discussing code.	Code Llama	Provided mo...
<input type="checkbox"/> Retrieval-Augmented Generation (15)		<input type="checkbox"/> Mistral AI (3)		flan-t5-xl-3b	A pretrained T5 - an encoder-decoder model pre-trained on a mixture of supervised / unsupervised tasks convert...	Google	Provided mo...
<input type="checkbox"/> Classification (19)		<input type="checkbox"/> IBM (8)		flan-t5-xxl-11b	flan-t5-xxl is an 11 billion parameter model based on the Flan-T5 family.	Google	Provided mo...
<input type="checkbox"/> Generation (19)		<input type="checkbox"/> Meta (4)		flan-ul2-20b	flan-ul2 is an encoder decoder model based on the T5 architecture and instruction-tuned using the Fine-tuned...	Google	Provided mo...
<input type="checkbox"/> Code generation and conversion (8)			merlinitite-7b	Merlinitite-7b is a Mistral-7b-derivative model trained with the LAB methodology, using Mixtral-8x7b-Instruct as a teacher...	Mistral AI, tuned ...	InstructLab	
<input type="checkbox"/> Extraction (19)			mixtral-8x7b-instruct-v0...	Mixtral-8-7b-instruct-v01-gptq model is made with AutoGPTQ, which mainly leverages the quantization technique to...	Mistral AI, tuned ...	Provided mo...	
			granite-13b-instruct-v2	The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for...	IBM	Provided mo...	
			granite-20b-code-instruct	The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for...			
			granite-20b-multilingual	The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for...			
			granite-34b-code-instruct	The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for...			

CONTINUE
@employees.pgm.sqlrpgle write a new procedure to compute the fibonacci sequence. Include only the new procedure in the response
watsonx - Mistral @codebase

> 1 context item
Certainly! Below is the new procedure to compute the Fibonacci sequence. You can add this procedure to your existing employees.pgm.sqlrpgle file.

```
// Procedure to compute Fibonacci sequence
Dcl-Proc ComputeFibonacci;
  Dcl-S n Int(5) Inz(10); // Number of terms to compute
  Dcl-S fibArray Int(10) Dim(100); // Array to store Fibonacci numbers
  Dcl-S i Int(5);

  // Initialize the first two terms
  fibArray(1) = 0;
  fibArray(2) = 1;

  // Compute the Fibonacci sequence
  For i = 3 To n;
    fibArray(i) = fibArray(i - 1) + fibArray(i - 2);
  EndFor;

  // Display the Fibonacci sequence
  For i = 1 To n;
    DSPLY ('Fibonacci(' + %Char(i) + ') = ' + %Char(fibArray(i)));
  EndFor;
End-Proc;
```

You can call the ComputeFibonacci procedure from anywhere in your program to compute and display the Fibonacci sequence. For example, you can call it in the main program logic like this:

```
// Call the Fibonacci procedure
ComputeFibonacci();
```

```
employees.pgm.sqlrpgle 9+, M ● {} config.json 3 depts.pgm.sqlrpgle mypgm.pgm.rpgle
qrpglesrc > employees.pgm.sqlrpgle > ...
123 Dcl-Proc HandleInputs;
128   If (%EOF(emps));
129     Iter;
130   Endif;
131
132   SelVal = %Trim(XSEL);
133
134   Select;
135     When (SelVal = '5');
136       DSPLY XID;
137     Ends1;
138
139   If (XSEL <> *Blank);
140     XSEL = *Blank;
141     Update SFLDTA;
142     SFLRRN = rrrn; Variable name casing does not match definition.
143   Endif;
144 Enddo;
145 End-Proc;
146
```

PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, **/*.ts, !**/node_modules/**)

- {} config.json C:\Users\911879897\continue 3
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cla... (1) [Ln 6, Col 16]
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cl... (1) [Ln 11, Col 16]
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cl... (1) [Ln 21, Col 16]
- employees.pgm.sqlrpgle qrpglesrc 16
 - Incorrect indentation. Expected 0, got 6 [Ln 8, Col 1]
 - Comments must be correctly formatted. [Ln 8, Col 7]
 - Incorrect indentation. Expected 0, got 6 [Ln 12, Col 11]


```

CONTINUE
// Procedure to compute Fibonacci sequence
Dcl-Proc ComputeFibonacci;
  Dcl-S n Int(5) Inz(10); // Number of terms to compute
  Dcl-S fibArray Int(10) Dim(100); // Array to store Fibonacci terms
  Dcl-S i Int(5);

  // Initialize the first two terms
  fibArray(1) = 0;
  fibArray(2) = 1;

  // Compute the Fibonacci sequence
  For i = 3 To n;
    fibArray(i) = fibArray(i - 1) + fibArray(i - 2);
  EndFor;

  // Display the Fibonacci sequence
  For i = 1 To n;
    DSPLY ('Fibonacci(' + %Char(i) + ') = ' + %Char(fibArray(i)));
  EndFor;
End-Proc;

```

You can call the `ComputeFibonacci` procedure from anywhere in your program to compute and display the Fibonacci sequence. For example, you can call it in the main program logic like this:

```
// Call the Fibonacci procedure
ComputeFibonacci();
```

This will compute and display the first 10 terms of the Fibonacci sequence. You can adjust the value of `n` in `employees.pgm.sqlrpgle` to compute a different number of terms.

Please write unit tests for this procedure

watsonx - Mistral @codebase

```

employees.pgm.sqlrpgle 9+, M • {} config.json 3 depts.pgm.sqlrpgle mypgm.pgm.rpgle
qrpglesrc > employees.pgm.sqlrpgle > ...
123 Dcl-Proc HandleInputs;
128   It (%EOF(emps));
129   Iter;
130   Endif;
131
132   SelVal = %Trim(XSEL);
133
134   Select;
135     When (SelVal = '5');
136       DSPLY XID;
137     Ends1;
138
139   If (XSEL <> *Blank);
140     XSEL = *Blank;
141     Update SFLDTA;
142     SFLRRN = rrrn; Variable name casing does not match definition.
143   Endif;
144 Enddo;
145 End-Proc;
146

```

PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, **/*.ts, !**/node_modules/**)

- {} config.json C:\Users\911879897\continue 3
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cla... (1) [Ln 6, Col 16]
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cl... (1) [Ln 11, Col 16]
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cl... (1) [Ln 21, Col 16]
- employees.pgm.sqlrpgle qrpglesrc 16
 - Incorrect indentation. Expected 0, got 6 [Ln 8, Col 1]
 - Comments must be correctly formatted. [Ln 8, Col 7]
 - Incorrect indentation. Expected 0, got 6 [Ln 12, Col 11]

```

CONTINUE
// Unit test procedure
Dcl-Proc TestComputeFibonacci;
  Dcl-S n Int(5) Inz(10); // Number of terms to compute
  Dcl-S fibArray Int(10) Dim(100); // Array to store Fibonacci numbers
  Dcl-S expectedFibArray Int(10) Dim(100); // Expected Fibonacci numbers
  Dcl-S i Int(5);
  Dcl-S testPassed Ind Inz(*On);

  // Initialize the expected Fibonacci sequence
  expectedFibArray(1) = 0;
  expectedFibArray(2) = 1;
  expectedFibArray(3) = 1;
  expectedFibArray(4) = 2;
  expectedFibArray(5) = 3;
  expectedFibArray(6) = 5;
  expectedFibArray(7) = 8;
  expectedFibArray(8) = 13;
  expectedFibArray(9) = 21;
  expectedFibArray(10) = 34;

  // Call the ComputeFibonacci procedure
  ComputeFibonacci();

  // Verify the output
  For i = 1 To n;
    If (fibArray(i) <> expectedFibArray(i));
      DSPLY ('Test failed at Fibonacci(' + %Char(i) + ')');
      testPassed = *Off;
    EndIf;
  EndFor;

  If (testPassed);
    DSPLY ('All tests passed!');
  Else;
    DSPLY ('Some tests failed.');
```

^ Cancel

```

employees.pgm.sqlrpgle 9+, M • {} config.json 3 depts.pgm.sqlrpgle mypgm.pgm.rpgle
qrpglesrc > employees.pgm.sqlrpgle > ...
123 Dcl-Proc HandleInputs;
128   It (%EOF(emps));
129   Iter;
130   Endif;
131
132   SelVal = %Trim(XSEL);
133
134   Select;
135     When (SelVal = '5');
136       DSPLY XID;
137   Ends1;
138
139   If (XSEL <> *Blank);
140     XSEL = *Blank;
141     Update SFLDTA;
142     SFLRRN = rrrn; Variable name casing does not match definition.
143   Endif;
144 Enddo;
145 End-Proc;
146
```

PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, **/*.ts, !**/node_modules/**)

- {} config.json C:\Users\911879897\continue 3
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cla... (1) [Ln 6, Col 16]
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cl... (1) [Ln 11, Col 16]
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cl... (1) [Ln 21, Col 16]
- employees.pgm.sqlrpgle qrpglesrc 16
 - Incorrect indentation. Expected 0, got 6 [Ln 8, Col 1]
 - Comments must be correctly formatted. [Ln 8, Col 7]
 - Incorrect indentation. Expected 0, got 6 [Ln 12, Col 11]

Ln 146, Col 1 Spaces: 2 UTF-8 LF RPGLE ✓ Continue


```
CONTINUE
If (testPassed);
  DSPLY ('All tests passed!');
Else;
  DSPLY ('Some tests failed.');
```

Explanation:

- ComputeFibonacci Procedure:** This is the procedure that computes the Fibonacci sequence and displays it.
- TestComputeFibonacci Procedure:** This is the unit test procedure that:
 - Initializes the expected Fibonacci sequence.
 - Calls the `ComputeFibonacci` procedure.
 - Verifies the output by comparing it with the expected sequence.
 - Displays the result of the test.

Running the Tests:

To run the tests, simply compile and run the program. The `TestComputeFibonacci` procedure will be called, and it will display whether the tests passed or failed.

This approach provides a basic framework for unit testing in RPGLE. For more advanced testing, you might consider using external tools or frameworks that can integrate with RPGLE.

Ask a follow-up
watsonx - Mistral

```
employees.pgm.sqlrpgle 9+, M • {} config.json 3 depts.pgm.sqlrpgle mypgm.pgm.rpgle
qrpglesrc > employees.pgm.sqlrpgle > ...
123 Dcl-Proc HandleInputs;
128   It (%EOF(emps));
129   Iter;
130   Endif;
131
132   SelVal = %Trim(XSEL);
133
134   Select;
135     When (SelVal = '5');
136       DSPLY XID;
137   Ends1;
138
139   If (XSEL <> *Blank);
140     XSEL = *Blank;
141     Update SFLDTA;
142     SFLRRN = rrrn; Variable name casing does not match definition.
143   Endif;
144   Enddo;
145 End-Proc;
146
```

PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, **/*.ts, !**/node_modules/**)

- {} config.json C:\Users\911879897\continue 3
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cla... (1) [Ln 6, Col 16]
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cl... (1) [Ln 11, Col 16]
 - Value is not accepted. Valid values: "gpt-4o", "codestral-latest", "llama3.1-70b", "llama3.1-405b", "gpt-3.5-turbo", "gemini-pro", "claude-3-5-sonnet-20240620", "cl... (1) [Ln 21, Col 16]
- employees.pgm.sqlrpgle qrpglesrc 16
 - Incorrect indentation. Expected 0, got 6 [Ln 8, Col 1]
 - Comments must be correctly formatted. [Ln 8, Col 7]
 - Incorrect indentation. Expected 0, got 6 [Ln 12, Col 11]

CONTINUE
the fibonacci sequence. Include only the new procedure in the response
watsonx - Mistral @codebase

> 1 context item

Certainly! Below is the new procedure to compute the Fibonacci sequence. You can add this procedure to your existing `employees.pgm.sqlrpgle` file.

```
// Procedure to compute Fibonacci sequence
Dcl-Proc ComputeFibonacci;
  Dcl-S n Int(5) Inz(10); // Number of terms to compute
  Dcl-S fibArray Int(10) Dim(100); // Array to store Fibonacci sequence
  Dcl-S i Int(5);

  // Initialize the first two terms
  fibArray(1) = 0;
  fibArray(2) = 1;

  // Compute the Fibonacci sequence
  For i = 3 To n;
    fibArray(i) = fibArray(i - 1) + fibArray(i - 2);
  EndFor;

  // Display the Fibonacci sequence
  For i = 1 To n;
    DSPLY ('Fibonacci(' + %Char(i) + ') = ' + %Char(fibArray(i)));
  EndFor;
End-Proc;
```

You can call the `ComputeFibonacci` procedure from anywhere in your program to compute and display the Fibonacci sequence. For example, you can call it in the main program logic like this:

```
// Call the Fibonacci procedure
ComputeFibonacci();
```

```
employees.pgm.sqlrpgle 9+, M
qrpglesrc > employees.pgm.sqlrpgle > ...
123 Dcl-Proc HandleInputs;
143   Endif;
144   Enddo;
145 End-Proc;
146 // Procedure to compute Fibonacci sequence
147 Dcl-Proc ComputeFibonacci;
148   Dcl-S n Int(5) Inz(10); // Number of terms to compute
149   Dcl-S fibArray Int(10) Dim(100); // Array to store Fibonacci sequence
150   Dcl-S i Int(5);
151
152   // Initialize the first two terms
153   fibArray(1) = 0;
154   fibArray(2) = 1;
155
156   // Compute the Fibonacci sequence
157   For i = 3 To n;
158     fibArray(i) = fibArray(i - 1) + fibArray(i - 2);
159   EndFor;
160
161   // Display the Fibonacci sequence
162   For i = 1 To n;
163     DSPLY ('Fibonacci(' + %Char(i) + ') = ' + %Char(fibArray(i)));
164   EndFor;
165 End-Proc;
166
```

- employees.pgm.sqlrpgle qrpglesrc 16
 - Incorrect indentation. Expected 0, got 6 [Ln 8, Col 1]
 - Comments must be correctly formatted. [Ln 8, Col 7]
 - Incorrect indentation. Expected 0, got 6 [Ln 12, Col 1]
 - Comments must be correctly formatted. [Ln 12, Col 7]
 - Variable name casing does not match definition. [Ln 14, Col 60]
 - Incorrect indentation. Expected 0, got 6 [Ln 38, Col 1]
 - Comments must be correctly formatted. [Ln 38, Col 7]


```

118     Sfldsp = *On;
119     SFLRRN = 1;
120     Endif;
121 End-Proc;
122
123 Dcl-Proc HandleInputs;
124   Dcl-p SelVal Char(1);
125
126   Dou (%EOF(emps));
127     ReadC SFLDTA;
128     If (%EOF(emps));
129       Iter;
130     Endif;
131
132     SelVal = %Trim(XSEL);
133
134     Select;
135       When (SelVal = '5');
136         DsPLY XID;
137     Ends1;
138
139     If (XSEL <> *Blank);
140       XSEL = *Blank;
141       Update SFLDTA;
142       SFLRRN = rrn; Variable name casing does not match definition.
143     Endif;
144   Enddo;
145 End-Proc;
146

```

```

employees.pgm.sqlrpgle (123-145)
Dcl-Proc HandleInputs;
  Dcl-p SelVal Char(1);

  Dou (%EOF(emps));
  ReadC SFLDTA;
  If (%EOF(emps));
    Iter;
  Endif;

  SelVal = %Trim(XSEL);

  Select;
  When (SelVal = '5');
    DsPLY XID;
  Ends1;

  If (XSEL <> *Blank);
    XSEL = *Blank;

```

Explain the above code in a few sentences without going into detail on specific methods.

The provided code is a procedure named `HandleInputs` written in SQLRPGLE. It processes input from a subfile named `SFLDTA` until the end of the file (`%EOF(emps)`) is reached. For each record, it reads the subfile data and trims the selection value (`XSEL`). If the selection value is '5', it displays the `XID`. If `XSEL` is not blank, it resets `XSEL` to blank, updates the subfile data, and sets the subfile record number (`SFLRRN`) to the current record number (`rrn`).

be more verbose

```

117 11 ( * ) ;
118     Sfldsp = *On;
119     SFLRRN = 1;
120     Endif;
121 End-Proc;
122
123 Dcl-Proc HandleInputs;
124   Dcl-p SelVal Char(1);
125
126   Dou (%EOF(emps));
127     ReadC SFLDTA;
128     If (%EOF(emps));
129       Iter;
130     Endif;
131
132     SelVal = %Trim(XSEL);
133
134     Select;
135       When (SelVal = '5');
136         DsPLY XID;
137     Ends1;
138
139     If (XSEL <> *Blank);
140       XSEL = *Blank;
141       Update SFLDTA;
142       SFLRRN = rrn; Variable name casing does not match definition.
143     Endif;
144   Enddo;
145 End-Proc;
146

```

qrpglesrc > employees.pgm.sqlrpgle > ...

118 Sfldsp = *On;
119 SFLRRN = 1;
120 Endif;
121 End-Proc;
122

Explain | Docstring

123 Dcl-Proc HandleInputs;
124 Dcl-p SelVal Char(1);

125
126 Dou (%EOF(emps));
127 ReadC SFLDTA;
128 If (%EOF(emps));
129 Iter;
130 Endif;

131
132 SelVal = %Trim(XSEL);

133
134 Select;
135 When (SelVal = '5');
136 DsPLY XID;
137 Ends1;

138
139 If (XSEL <> *Blank);
140 XSEL = *Blank;
141 Update SFLDTA;
142 SFLRRN = rrn; Variable name casing does not match definition.

143 Endif;
144 Enddo;
145 End-Proc;

146

117 11 (*) ;
118 Sfldsp = *On;
119 SFLRRN = 1;
120 Endif;
121 End-Proc;
122
123 Dcl-Proc HandleInputs;
124 Dcl-p SelVal Char(1);
125
126 Dou (%EOF(emps));
127 ReadC SFLDTA;
128 If (%EOF(emps));
129 Iter;
130 Endif;
131
132 SelVal = %Trim(XSEL);
133
134 Select;
135 When (SelVal = '5');
136 DsPLY XID;
137 Ends1;
138
139 If (XSEL <> *Blank);
140 XSEL = *Blank;
141 Update SFLDTA;
142 SFLRRN = rrn; Variable name casing does not match definition.
143 Endif;
144 Enddo;
145 End-Proc;
146

be more verbose
Mistral WatsonX + Add Context Alt Use active file Enter

Certainly! Let's break down the code in more detail:

The code defines a procedure named `HandleInputs` in SQLRPGLE. This procedure is designed to handle user inputs from a subfile named `SFLDTA`. Here's a step-by-step explanation:

1. Procedure Declaration:

```
Dcl-Proc HandleInputs;
```

This line declares the start of the `HandleInputs` procedure.

2. Variable Declaration:

```
Dcl-p SelVal Char(1);
```

This line declares a local variable `SelVal` of type character with a length of 1. This variable will be used to store the trimmed value of the selection input (`XSEL`).

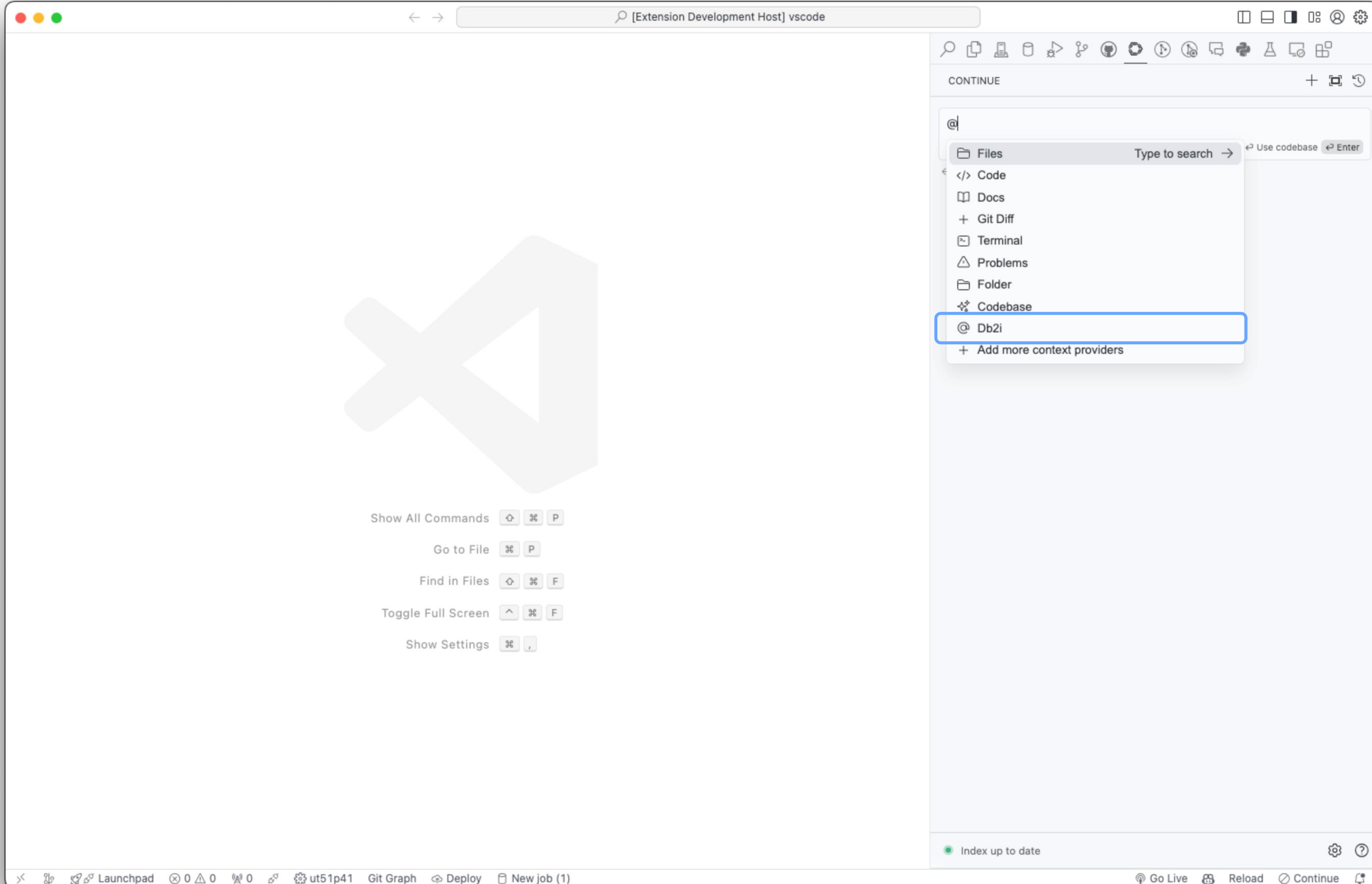
3. Loop Until End of File:

```
Dou (%EOF(emps));
```

This line starts a loop that continues until the end of the file (`%EOF(emps)`) is reached. The `%EOF(emps)` function checks if the end of the file has been reached.

4. Read Subfile Data:

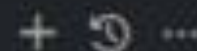
Db2 for IBMi context provider



- 1. Connect to an IBMi system using Code4i
- 2. invoke "Db2" context provider in continue extension



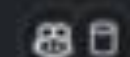
CHAT



inline chat session.

I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve. Check out the [Copilot documentation](#) to learn more.

@Db2i /activity



The system currently has a total of 246 jobs. The maximum number of jobs that can run in the system is 163,520. Out of the 246 jobs, 221 are currently active. There are no interactive jobs running in the system. Additionally, there are no jobs signed on, disconnected, suspended by sysreq, suspended by group job, or waiting to be printed.

In terms of batch jobs, 90 jobs are currently running, 1 is ending, and 9 are waiting to run. There are no batch jobs on a held job queue, held while running, suspended while running, or waiting for a message.

The system has a total of 66,510,848 bytes of main storage, with 209,976 bytes of system ASP storage and total auxiliary storage. Currently, 27.16% of the system ASP storage is being used. The temporary storage being used at the moment is 9,970 bytes, with a maximum temporary storage usage of 27,517 bytes.

The system's CPU usage is as follows: the current CPU capacity is 0.5, with an average CPU rate, average CPU utilization, minimum CPU utilization, and maximum CPU utilization all at 0. The SQL CPU utilization is not available.

Other system information includes the host name "UT24P60", partition ID 60 out of a total of 91 partitions, and 1,039 active threads in the system. The system is not in a restricted state and has a partition name of "ut24p60". The system has 176 physical processors, with 175 of them being in the shared processor pool. The maximum number of physical processors is set to 176.

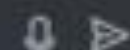
In terms of memory, the system has a defined memory of 65,536 bytes, with a minimum memory requirement of 32,768 bytes and a maximum memory capacity of 131,072 bytes. The memory increment is set to 256 bytes.

The system currently has 2 defined virtual processors and is utilizing all 2 virtual processors. The minimum and maximum virtual processors are set to 1 and 4, respectively. The defined processing capacity is 0.5, with the processing capacity currently being utilized at 0.5 as well. The minimum required processing capacity is 0.1, and the maximum licensed processing capacity is 176.

The system does not have any interactive capacity defined, nor is it currently utilizing any interactive capacity. The interactive threshold is set at 100.

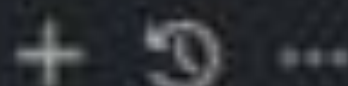
Other system configurations include a defined variable capacity weight of 128, a variable capacity weight

@Db2i Chat with the Db2 for i AI assistant





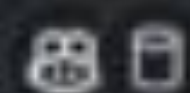
CHAT



inline chat session.

I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and [share feedback](#) so that we can learn and improve. Check out the [Copilot documentation](#) to learn more.

@Db2i /activity



The system currently has a total of 246 jobs. The maximum number of jobs that can run in the system is 163,520. Out of the 246 jobs, 221 are currently active. There are no interactive jobs running in the system. Additionally, there are no jobs signed on, disconnected, suspended by sysreq, suspended by group job, or waiting to be printed.

In terms of batch jobs, 90 jobs are currently running, 1 is ending, and 9 are waiting to run. There are no batch jobs on a held job queue, held while running, suspended while running, or waiting for a message.

The system has a total of 66,510,848 bytes of main storage, with 209,976 bytes of system ASP storage and total auxiliary storage. Currently, 27.16% of the system ASP storage is being used. The temporary storage being used at the moment is 9,970 bytes, with a maximum temporary storage usage of 27,517 bytes.

The system's CPU usage is as follows: the current CPU capacity is 0.5, with an average CPU rate, average CPU

Db2 for IBMi context provider: add additional context



The screenshot shows a VS Code interface with a SQL query in the editor, a chat window on the right, and a table of results in the bottom panel.

Editor: `SELECT * FROM SAMPLE.EMPLOYEE;`

Chat Window:

CONTINUE

@Db2i how can I fetch a list of all the employees using sql?

Ollama - ibm-granite-8b-code-instruct:latest + Add Context Use codebase Enter

Context Used

You can use the following SQL query to retrieve a list of all employees from the SAMPLE schema's EMPLOYEE table:

```
SELECT * FROM SAMPLE.EMPLOYEE;
```

This query will return all columns and rows in the table, including the employee number (EMPNO), first name (FIRSTNAME), middle initial (MIDINIT), last name (LASTNAME), work department (WORKDEPT), phone number (PHONENO), hire date (HIREDATE), job title (JOB), education level (EDLEVEL), sex (SEX), birthdate (BIRTHDATE), salary (SALARY), bonus (BONUS), and commission (COMM). Note that the query assumes that you are connected to the SAMPLE schema in your Db2i environment. If you are connected to a different schema, you will need to specify the correct schema name in place of "SAMPLE" in the query.

Ask a follow-up

Ollama - ibm-granite-8b-code-instruct:latest + Add Context Use codebase Enter

New Session (⌘ L)

Index up to date

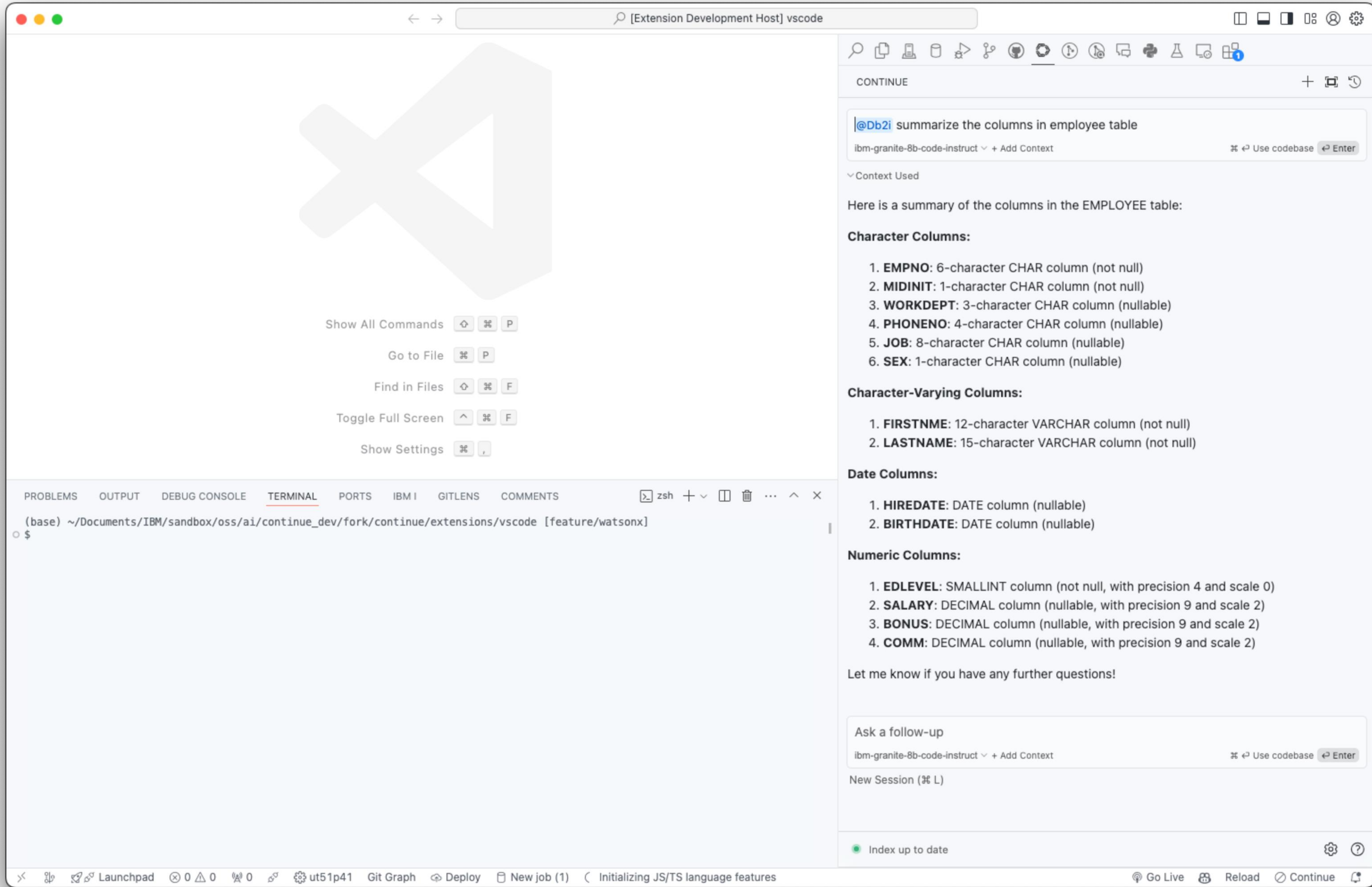
RESULTS:

EMPNO	FIRSTNAME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY
000010	CHRISTINE	I	HAAS	A00	3978	01/01/65	PRES	18	F	null	52700
000020	MICHAEL	L	THOMPSON	B01	3476	10/10/73	MANAGER	18	M	02/02/48	41500
000030	SALLY	A	KWAN	C01	4738	04/05/75	MANAGER	20	F	05/11/41	38000
000050	JOHN	B	GEYER	E01	6789	08/17/49	MANAGER	16	M	null	40000
000060	IRVING	F	STERN	D11	6423	09/14/73	MANAGER	16	M	07/07/45	32000
000070	EVA	D	PULASKI	D21	7831	09/30/80	MANAGER	16	F	05/26/53	36000
000090	EILEEN	W	HENDERSON	E11	5498	08/15/70	MANAGER	16	F	05/15/41	29000
000100	THEODORE	Q	SPENSER	E21	0972	06/19/80	MANAGER	14	M	12/18/56	26000
000110	VINCENZO	G	LUCCHESI	A00	3490	05/16/58	SALESREP	19	M	null	46000
000120	SEAN		O'CONNELL	A00	2167	12/05/63	CLERK	14	M	10/18/42	29000
000130	DELORES	M	QUINTANA	C01	4578	07/28/71	ANALYST	16	F	null	23000
000140	HEATHER	A	NICHOLLS	C01	1793	12/15/76	ANALYST	18	F	01/19/46	28000
000150	BRUCE		ADAMSON	D11	4510	02/12/72	DESIGNER	16	M	05/17/47	25000
000160	ELIZABETH	R	PIANKA	D11	3782	10/11/77	DESIGNER	17	F	04/12/55	22000
000170	MASATOSHI	J	YOSHIMURA	D11	2890	09/15/78	DESIGNER	16	M	01/05/51	24000
000180	MARILYN	S	SCOUTTEN	D11	1682	07/07/73	DESIGNER	17	F	02/21/49	21000
000190	JAMES	H	WALKER	D11	2986	07/26/74	DESIGNER	16	M	06/25/52	20000
000200	DAVID		BROWN	D11	4501	03/03/66	DESIGNER	16	M	05/29/41	27000

Automatically filter tables references

The context provide provides schema context

Db2 for IBMi context provider: table insights



The screenshot shows the VS Code interface with a chat window on the right. The chat window is titled "CONTINUE" and contains a prompt: "@Db2i summarize the columns in employee table". Below the prompt, it shows the context used: "ibm-granite-8b-code-instruct". The chat response provides a summary of the columns in the EMPLOYEE table, categorized into four groups:

- Character Columns:**
 1. **EMPNO:** 6-character CHAR column (not null)
 2. **MIDINIT:** 1-character CHAR column (not null)
 3. **WORKDEPT:** 3-character CHAR column (nullable)
 4. **PHONENO:** 4-character CHAR column (nullable)
 5. **JOB:** 8-character CHAR column (nullable)
 6. **SEX:** 1-character CHAR column (nullable)
- Character-Varying Columns:**
 1. **FIRSTNAME:** 12-character VARCHAR column (not null)
 2. **LASTNAME:** 15-character VARCHAR column (not null)
- Date Columns:**
 1. **HIREDATE:** DATE column (nullable)
 2. **BIRTHDATE:** DATE column (nullable)
- Numeric Columns:**
 1. **EDLEVEL:** SMALLINT column (not null, with precision 4 and scale 0)
 2. **SALARY:** DECIMAL column (nullable, with precision 9 and scale 2)
 3. **BONUS:** DECIMAL column (nullable, with precision 9 and scale 2)
 4. **COMM:** DECIMAL column (nullable, with precision 9 and scale 2)

The chat window also includes a "New Session" button and a "Go Live" button. The terminal at the bottom shows the current directory: "(base) ~/Documents/IBM/sandbox/oss/ai/continue_dev/fork/continue/extensions/vscode [feature/watsonx]".

Db2 for IBMi context provider: generate SQL

The screenshot shows the VS Code interface with a code editor on the left containing a SQL query. A blue box highlights the query: `SELECT E.EMPNO, E.FIRSTNME, E.LASTNAME, D.DEPTNAME FROM EMPLOYEE E JOIN DEPARTMENT D ON E.WORKDEPT = D.DEPTNO;`. A blue bracket points from this code to the text "Run Statement directly in VSCode".

On the right, a chat window shows a user asking: "@Db2i how can I find the department name for each employee?". The chat response provides an example query: `SELECT E.EMPNO, E.FIRSTNME, E.LASTNAME, D.DEPTNAME FROM EMPLOYEE E JOIN DEPARTMENT D ON E.WORKDEPT = D.DEPTNO;`. A blue box highlights this query. A blue bracket points from the chat response to the text "Referenced schemas", which lists "New job (1)-SAMPLE-DEPARTMENT" and "New job (1)-SAMPLE-EMPLOYEE".

Below the chat, a "RESULTS" table displays the output of the query:

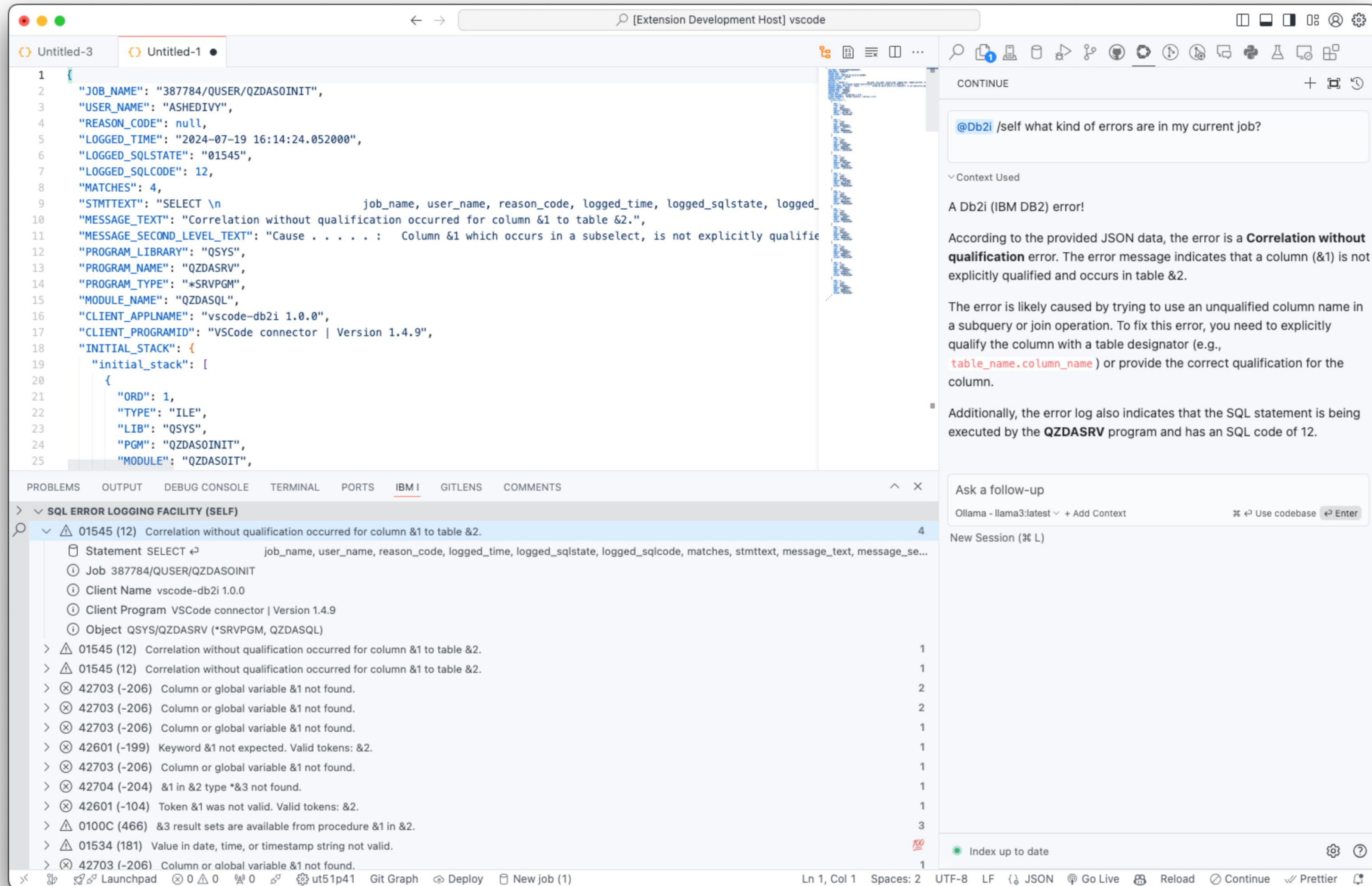
EMPNO	FIRSTNME	LASTNAME	DEPTNAME
000010	CHRISTINE	HAAS	SPIFFY COMPUTER SERVICE DIV.
000110	VINCENZO	LUCCHESSI	SPIFFY COMPUTER SERVICE DIV.
000120	SEAN	O'CONNELL	SPIFFY COMPUTER SERVICE DIV.
200010	DIAN	HEMMINGER	SPIFFY COMPUTER SERVICE DIV.
200120	GREG	ORLANDO	SPIFFY COMPUTER SERVICE DIV.
000020	MICHAEL	THOMPSON	PLANNING
000030	SALLY	KWAN	INFORMATION CENTER
000130	DELORES	QUINTANA	INFORMATION CENTER
000140	HEATHER	NICHOLLS	INFORMATION CENTER
200140	KIM	NATZ	INFORMATION CENTER
000060	IRVING	STERN	MANUFACTURING SYSTEMS
000150	BRUCE	ADAMSON	MANUFACTURING SYSTEMS
000160	ELIZABETH	PIANKA	MANUFACTURING SYSTEMS
000170	MASATOSHI	YOSHIMURA	MANUFACTURING SYSTEMS
000180	MARILYN	SCOUTTEN	MANUFACTURING SYSTEMS
000190	JAMES	WALKER	MANUFACTURING SYSTEMS
000200	DAVID	BROWN	MANUFACTURING SYSTEMS
000210	WILLIAM	JONES	MANUFACTURING SYSTEMS
000220	JENNIFER	LUTZ	MANUFACTURING SYSTEMS
200170	KIYOSHI	YAMAMOTO	MANUFACTURING SYSTEMS
200220	REBA	JOHN	MANUFACTURING SYSTEMS
000070	EVA	PULASKI	ADMINISTRATION SYSTEMS

At the bottom of the chat window, there is an "Ask a follow-up" section with a text input field and a "New Session (# L)" button.

Referenced schemas

Correctly join sample and department tables

Db2 for IBMi: Assistance with SELF Errors (future?)



The screenshot displays the VS Code editor with a JSON file containing Db2 error information. The error message is: "Correlation without qualification occurred for column &1 to table &2." The extension interface on the right shows a chat window with the prompt "@Db2i /self what kind of errors are in my current job?". The response explains the error as a "Correlation without qualification" error, likely caused by an unqualified column name in a subquery or join operation. The bottom panel shows the "SQL ERROR LOGGING FACILITY (SELF)" with a list of errors, including the one being discussed.

```
1 {
2   "JOB_NAME": "387784/QUSER/QZDASOINIT",
3   "USER_NAME": "ASHEDIVY",
4   "REASON_CODE": null,
5   "LOGGED_TIME": "2024-07-19 16:14:24.052000",
6   "LOGGED_SQLSTATE": "01545",
7   "LOGGED_SQLCODE": 12,
8   "MATCHES": 4,
9   "STMTTEXT": "SELECT \n
10  job_name, user_name, reason_code, logged_time, logged_sqlstate, logged_
11  \"MESSAGE_TEXT\": \"Correlation without qualification occurred for column &1 to table &2.\",
12  \"MESSAGE_SECOND_LEVEL_TEXT\": \"Cause . . . . . : Column &1 which occurs in a subselect, is not explicitly qualifie
13  \"PROGRAM_LIBRARY\": \"QSYS\",
14  \"PROGRAM_NAME\": \"QZDASRV\",
15  \"PROGRAM_TYPE\": \"*SRVPGM\",
16  \"MODULE_NAME\": \"QZDASQL\",
17  \"CLIENT_APPLNAME\": \"vscode-db2i 1.0.0\",
18  \"CLIENT_PROGRAMID\": \"VSCode connector | Version 1.4.9\",
19  \"INITIAL_STACK\": {
20    \"initial_stack\": [
21      {
22        \"ORD\": 1,
23        \"TYPE\": \"ILE\",
24        \"LIB\": \"QSYS\",
25        \"PGM\": \"QZDASOINIT\",
26        \"MODULE\": \"QZDASOIT\",
```

CONTINUE

@Db2i /self what kind of errors are in my current job?

Context Used

A Db2i (IBM DB2) error!

According to the provided JSON data, the error is a **Correlation without qualification** error. The error message indicates that a column (&1) is not explicitly qualified and occurs in table &2.

The error is likely caused by trying to use an unqualified column name in a subquery or join operation. To fix this error, you need to explicitly qualify the column with a table designator (e.g., `table_name.column_name`) or provide the correct qualification for the column.

Additionally, the error log also indicates that the SQL statement is being executed by the **QZDASRV** program and has an SQL code of 12.

Ask a follow-up

Ollama - llama3:latest + Add Context Use codebase Enter

New Session (% L)

Index up to date

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS **IBM I** GITLENS COMMENTS

SQL ERROR LOGGING FACILITY (SELF)

- 01545 (12) Correlation without qualification occurred for column &1 to table &2. 4
 - Statement SELECT job_name, user_name, reason_code, logged_time, logged_sqlstate, logged_sqlcode, matches, stmttext, message_text, message_se...
 - Job 387784/QUSER/QZDASOINIT
 - Client Name vscode-db2i 1.0.0
 - Client Program VSCode connector | Version 1.4.9
 - Object QSYS/QZDASRV (*SRVPGM, QZDASQL)
- 01545 (12) Correlation without qualification occurred for column &1 to table &2. 1
- 01545 (12) Correlation without qualification occurred for column &1 to table &2. 1
- 42703 (-206) Column or global variable &1 not found. 2
- 42703 (-206) Column or global variable &1 not found. 2
- 42703 (-206) Column or global variable &1 not found. 1
- 42601 (-199) Keyword &1 not expected. Valid tokens: &2. 1
- 42703 (-206) Column or global variable &1 not found. 1
- 42704 (-204) &1 in &2 type *&3 not found. 1
- 42601 (-104) Token &1 was not valid. Valid tokens: &2. 1
- 0100C (466) &3 result sets are available from procedure &1 in &2. 3
- 01534 (181) Value in date, time, or timestamp string not valid. 1
- 42703 (-206) Column or global variable &1 not found. 1

Ln 1, Col 1 Spaces: 2 UTF-8 LF JSON Go Live Reload Continue Prettier

Summarize the SELF Errors in current job

@Db2 /self will add the SELF Error data to prompt automatically

Future Watson Code Assistant for IBM i: Potential for advanced modernization activity

- Leverage community domain knowledge
- outperform other industry solutions
- Converting S/36 RPG, RPG II, etc. into Free-Format ILE RPG
- Converting monolithic to modular
- Suggesting sections of code where open source languages would help
- Transforming old data definitions and access into modern Db2 & SQL
- Helping modernize when other languages are the starting point
- ... and more!!!
- e-mail aiforibmi@ibm.com



Thank you!