

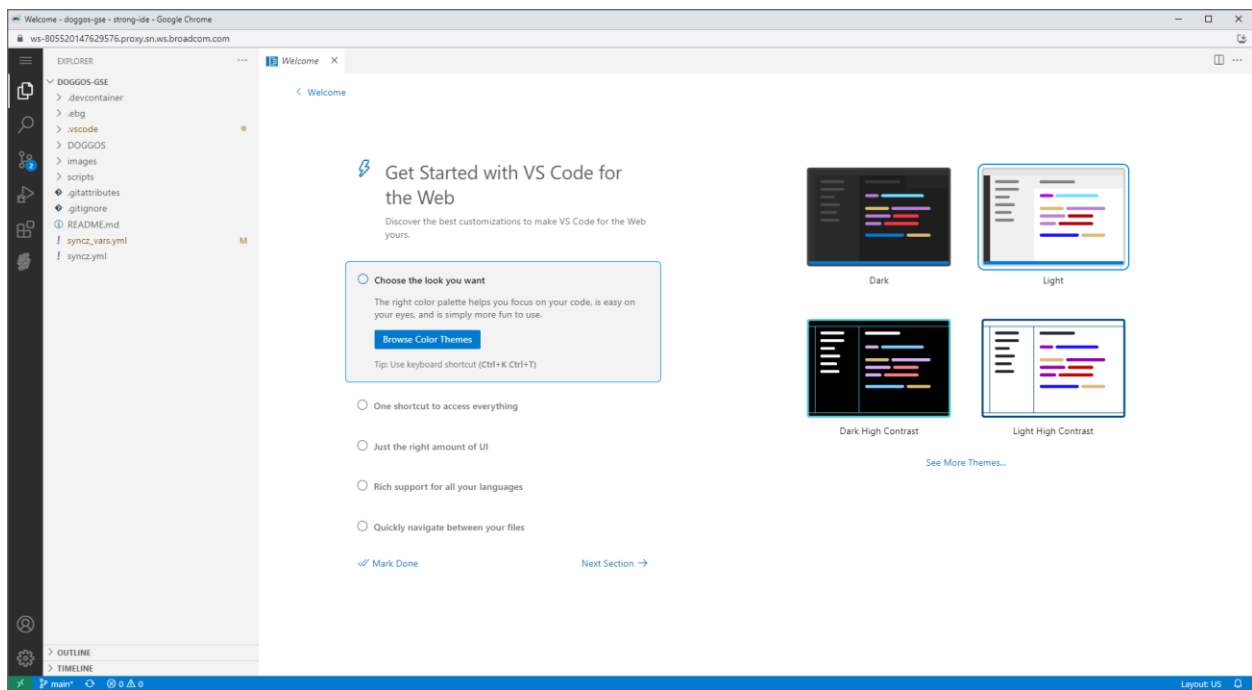
Dev/Ops Workshop – Dump Reading

In this lab exercise the student will perform the following tasks:

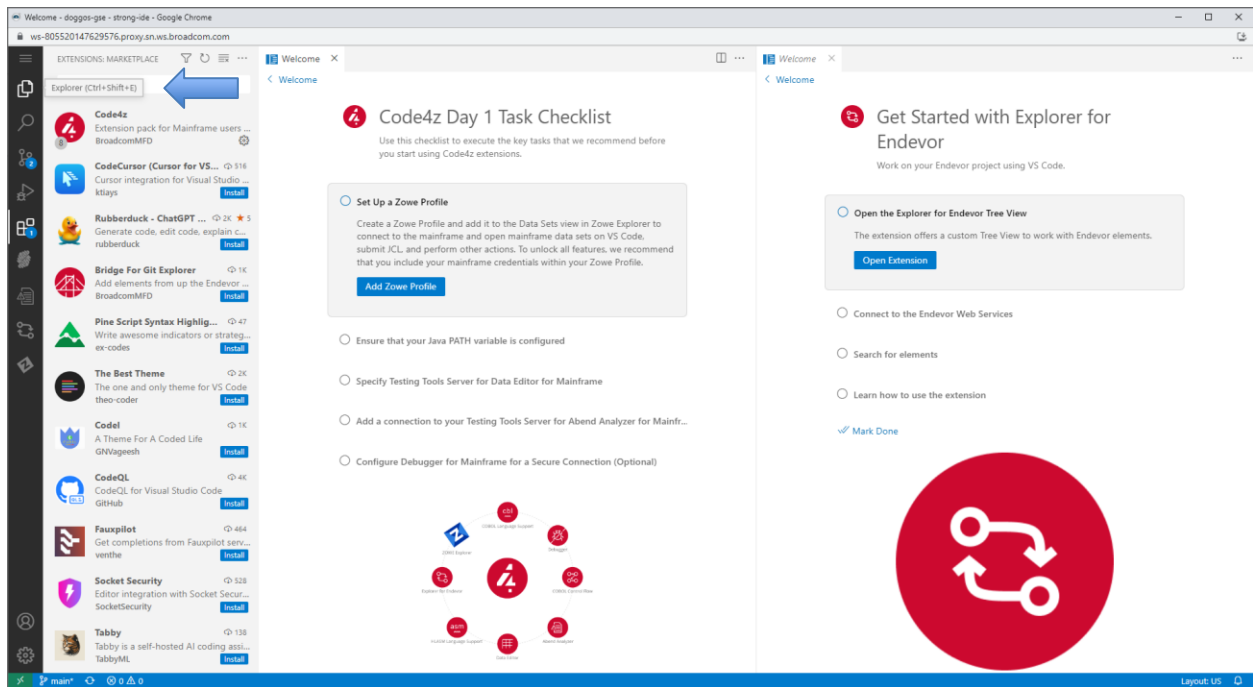
1. Sign on to an assigned workspace.
2. Review Abend Analyzer configuration.
3. Open a dump and shoot the problem.

Start up:

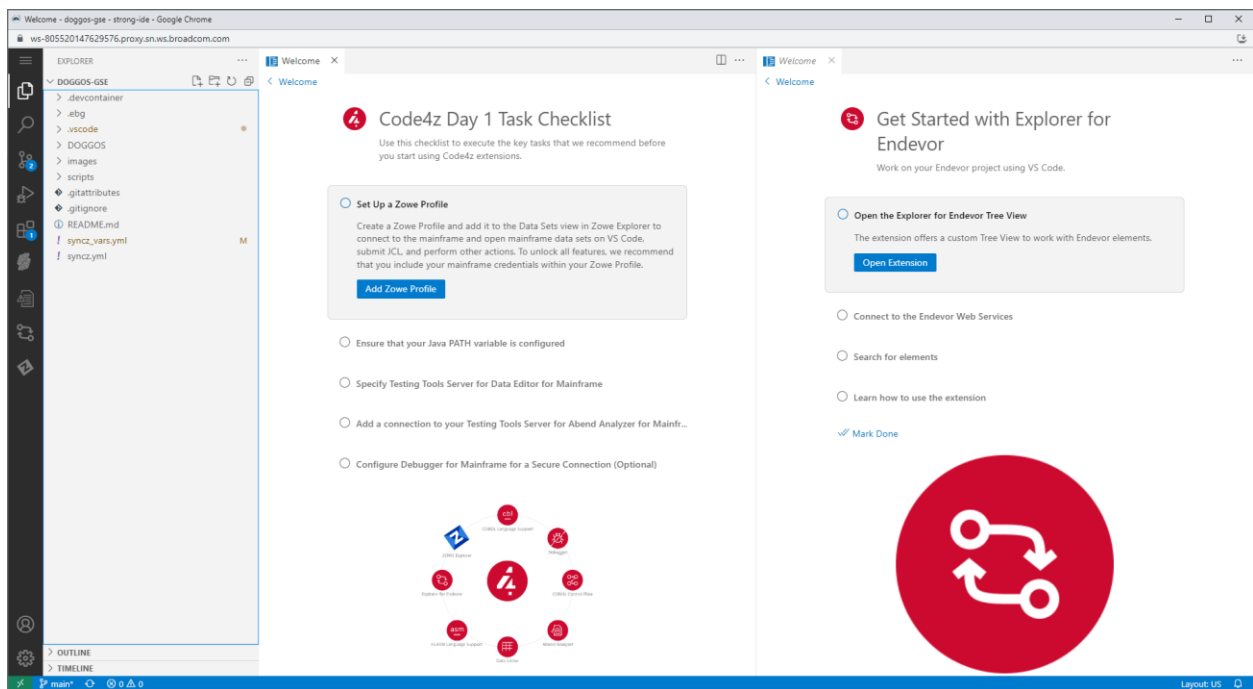
1. At this point you are now in your workspace in a VS Code session.



At this point you are ready to start the workshop. Make sure you are at the Explorer view click on the extension that looks like two pieces of paper.

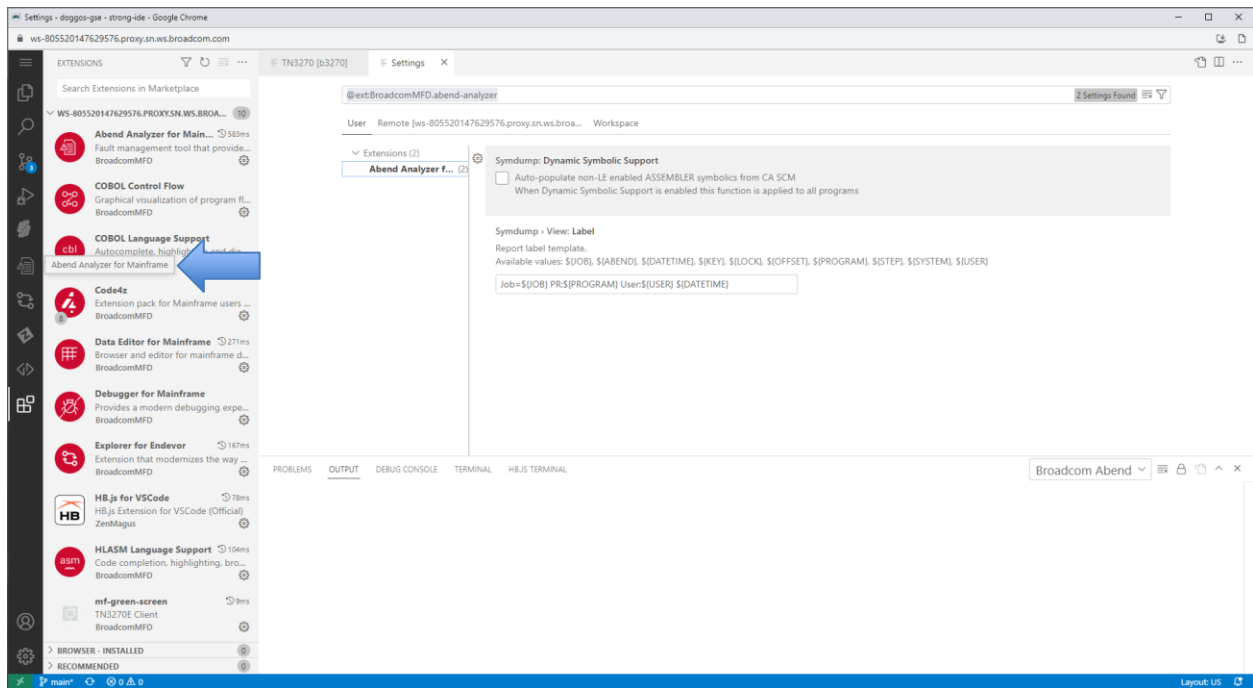


It will return you to the main view.

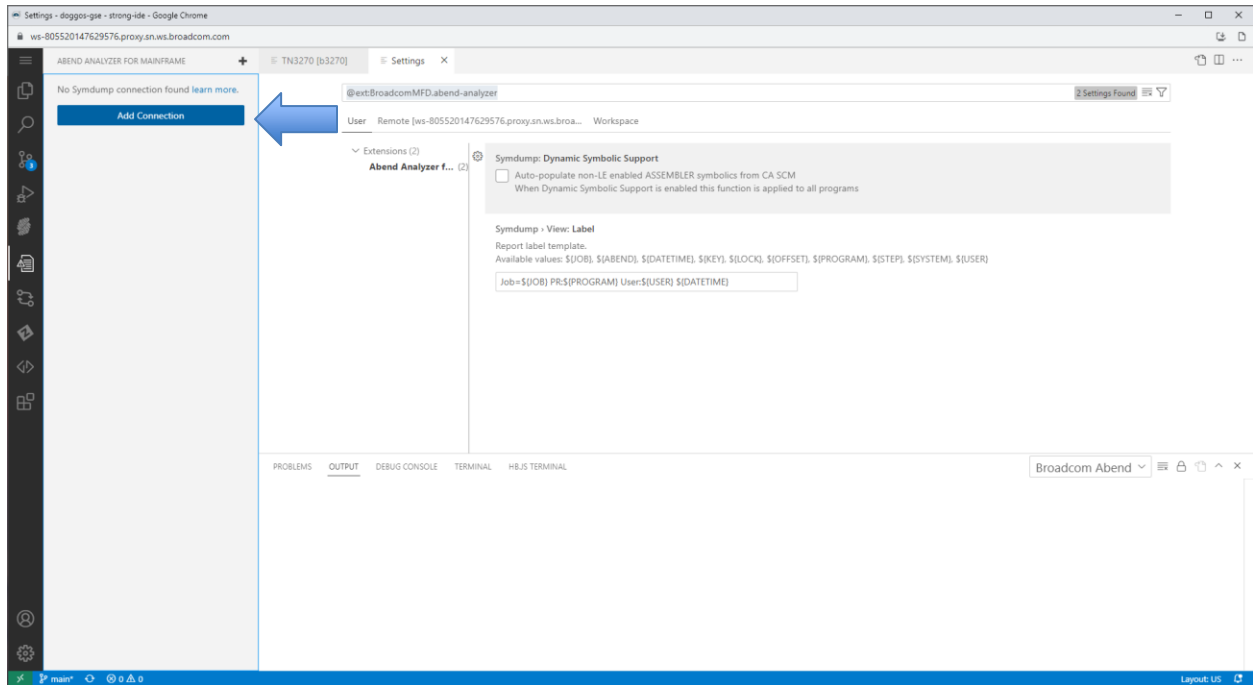


Configuring Abend Analyzer:

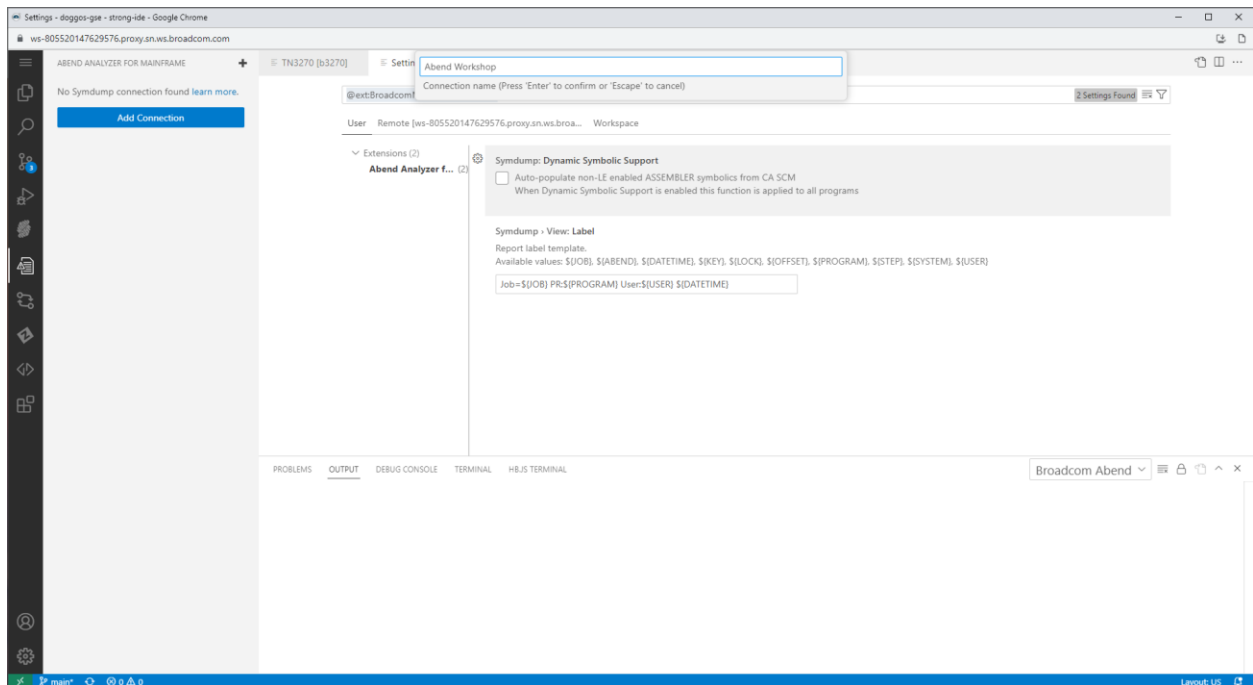
In this section the student will see the setup Abend Analyzer. This part has been completed already.



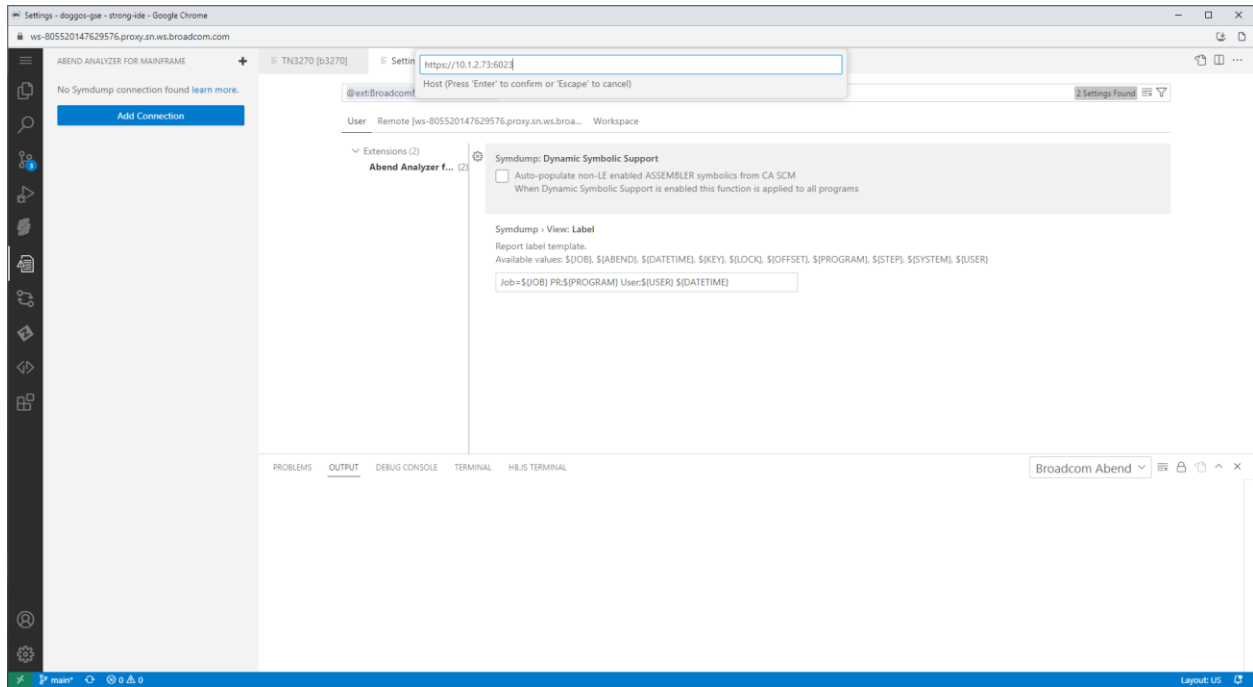
This will bring up screen that a mainframe connection can be added.



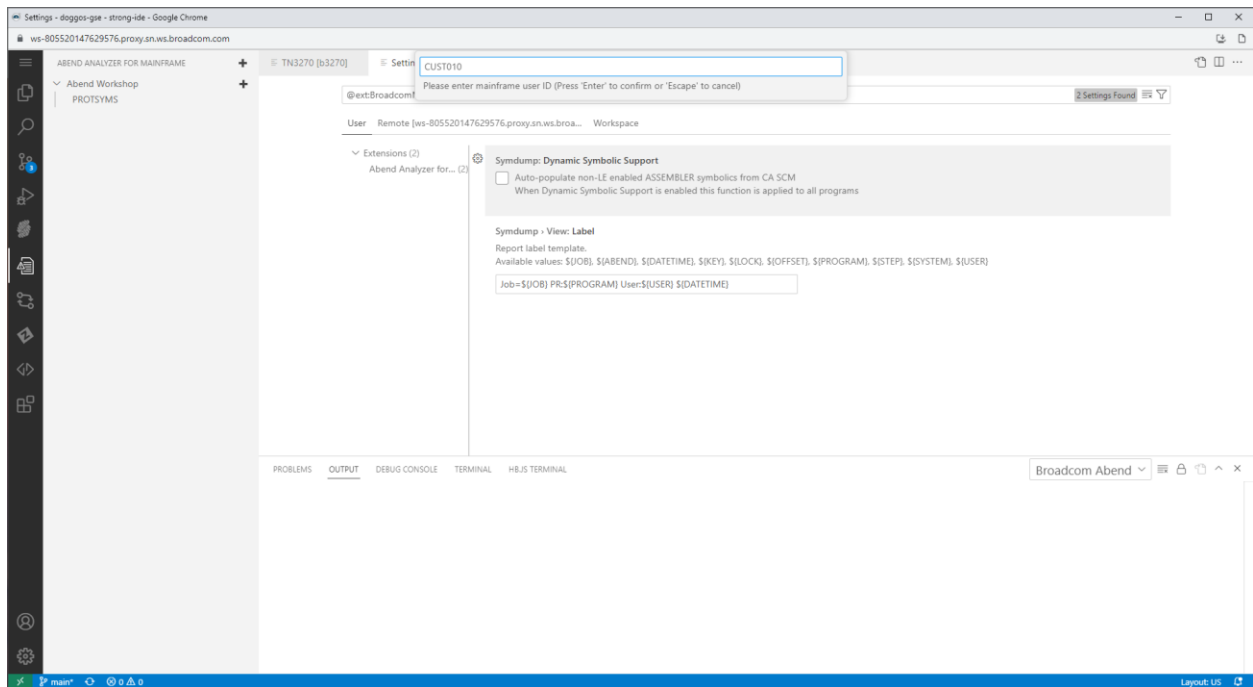
This will bring up a pop up box where the name of the connection has to be added. Provide any name (DEMO).



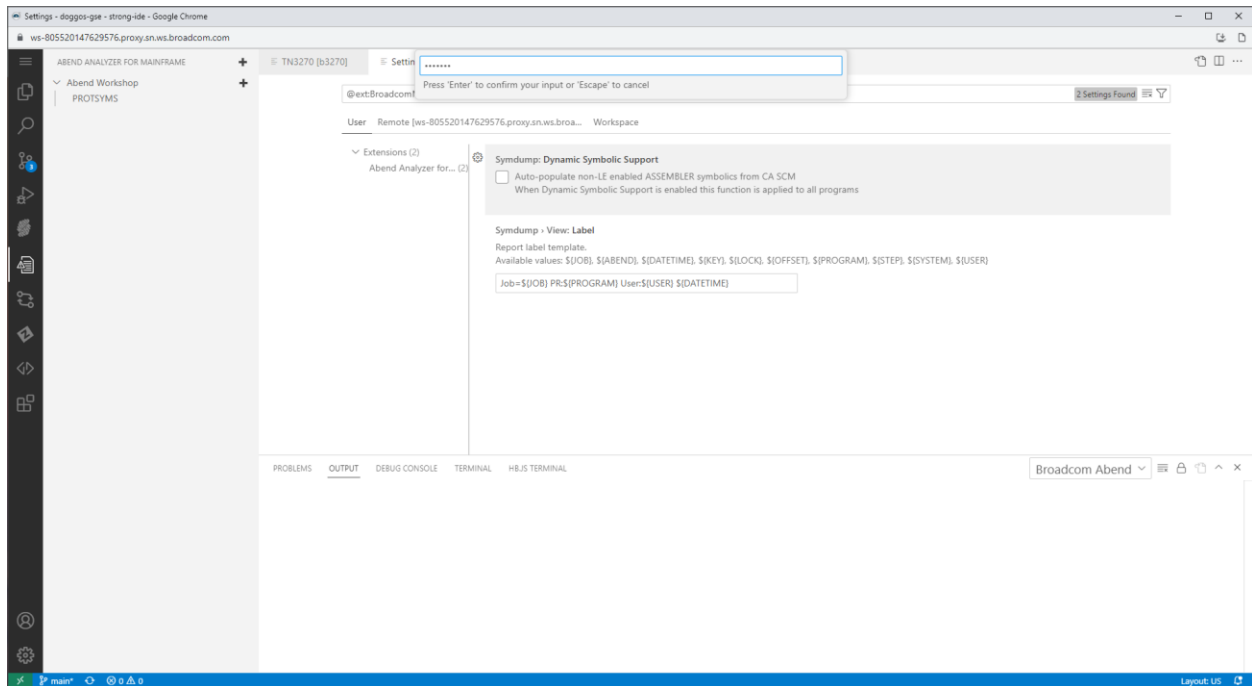
The next window that will pop up is the Host Name. For example
<https://10.1.2.73:6023>



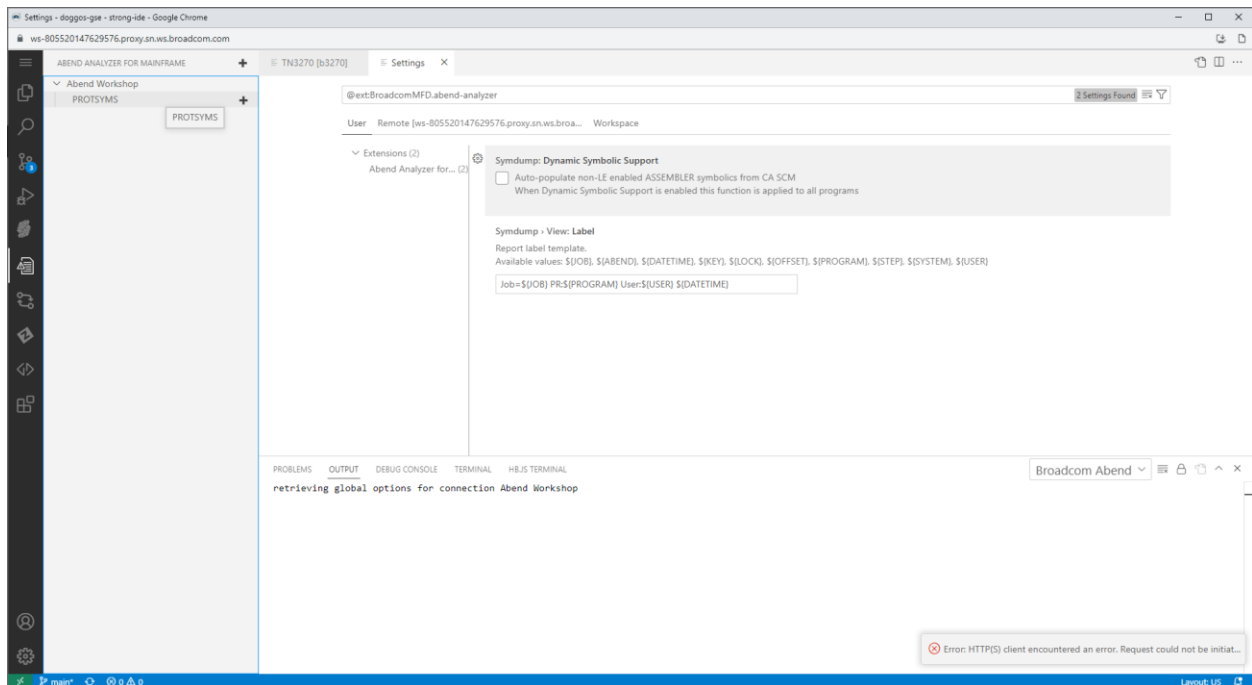
The next pop up will ask for a user ID. Your USERID was provided earlier. CUST0nn.



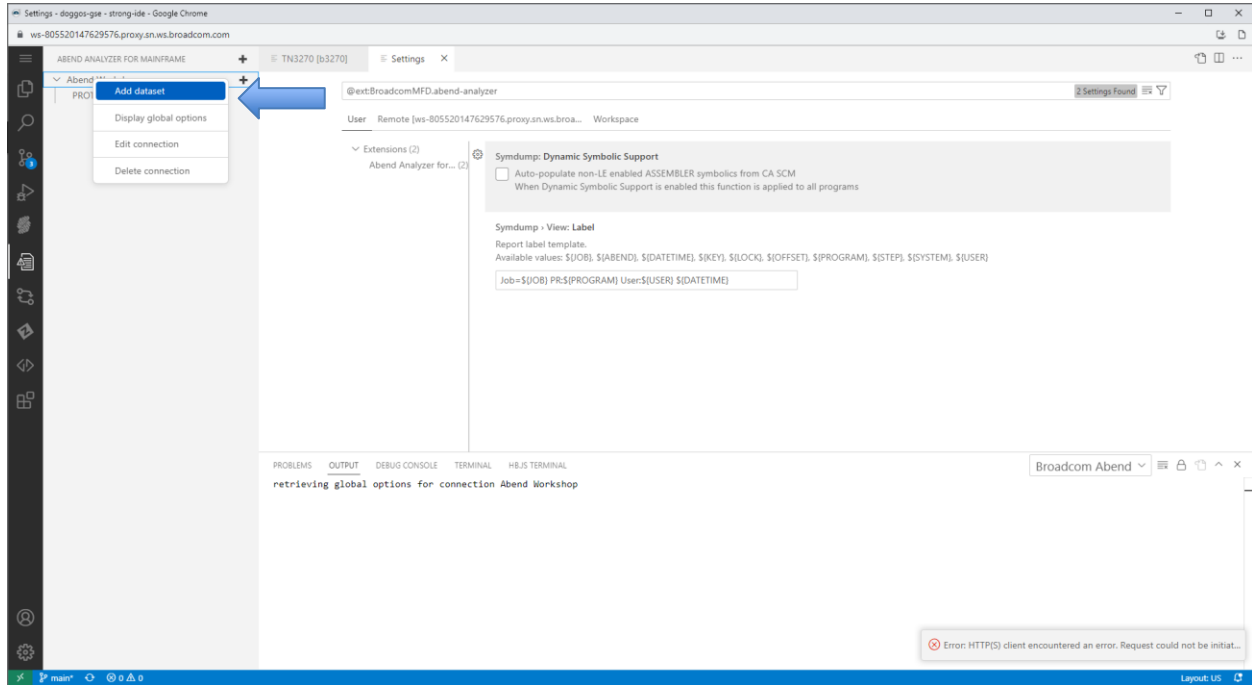
The next pop up will ask for a password. The Password is the same as the userid.



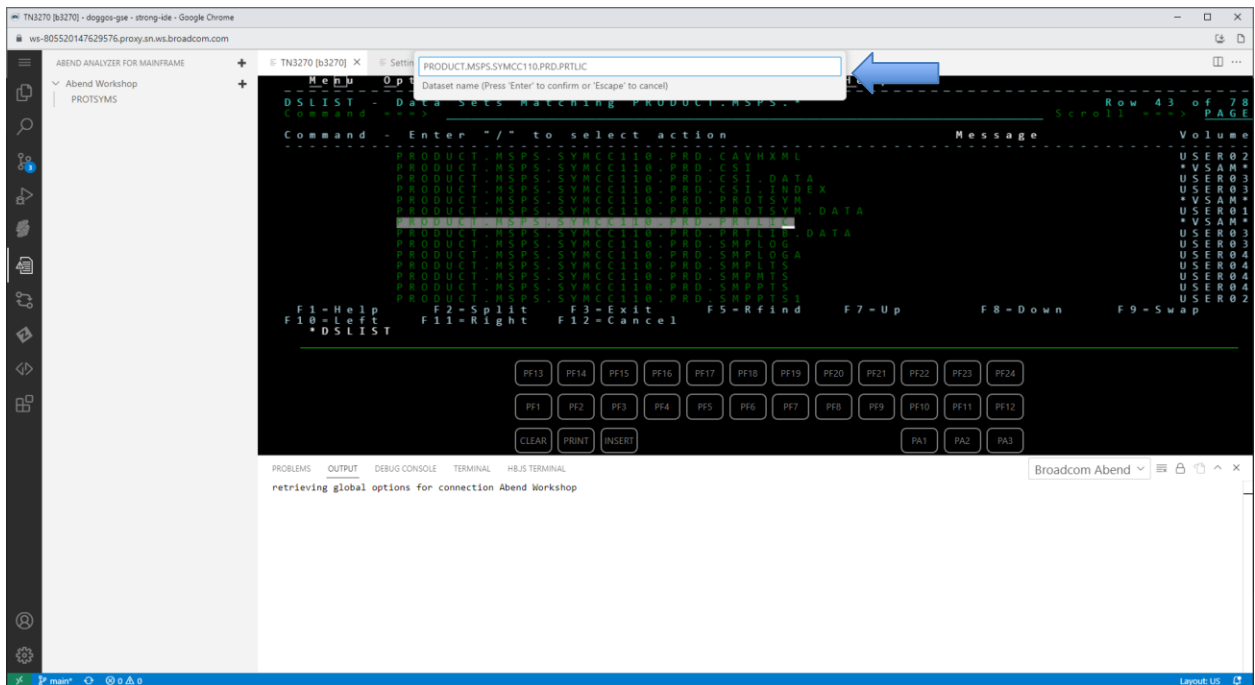
Notice now under ABEND ANALYZER FOR MAINFRAME you have the connection name and the PROTSYMS.



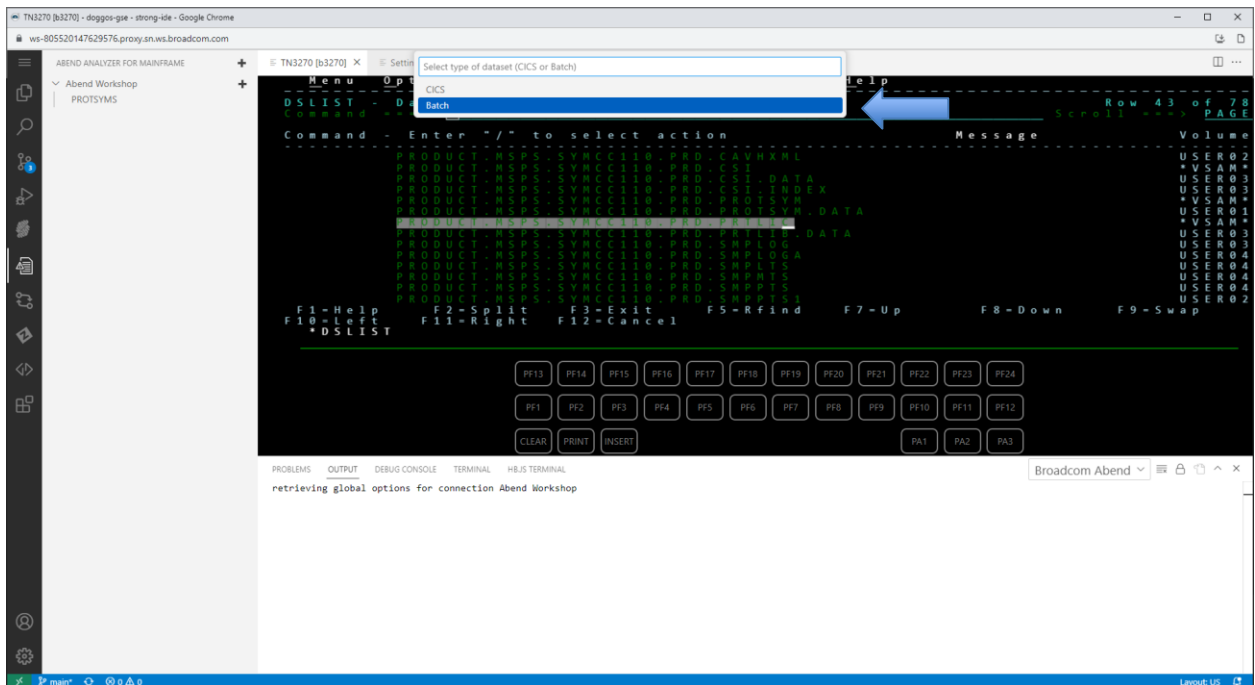
Now a prtlib dataset has to be added. The prtlib is where the dumps printouts are stored. To do this you would just right click on Abend Workshop and select Add dataset.



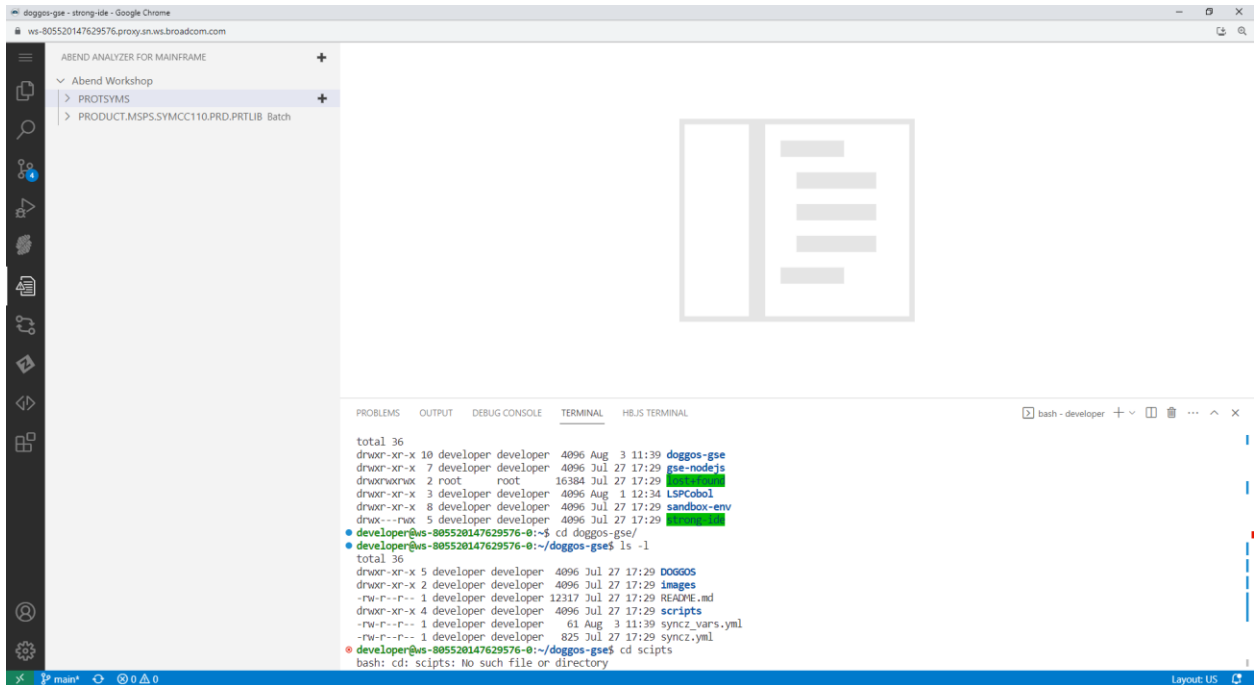
Add the dataset in the pop up box. The dataset name is as follows:
PRODUCT.MSPS.SYMCC110.PRD.PRTLIB



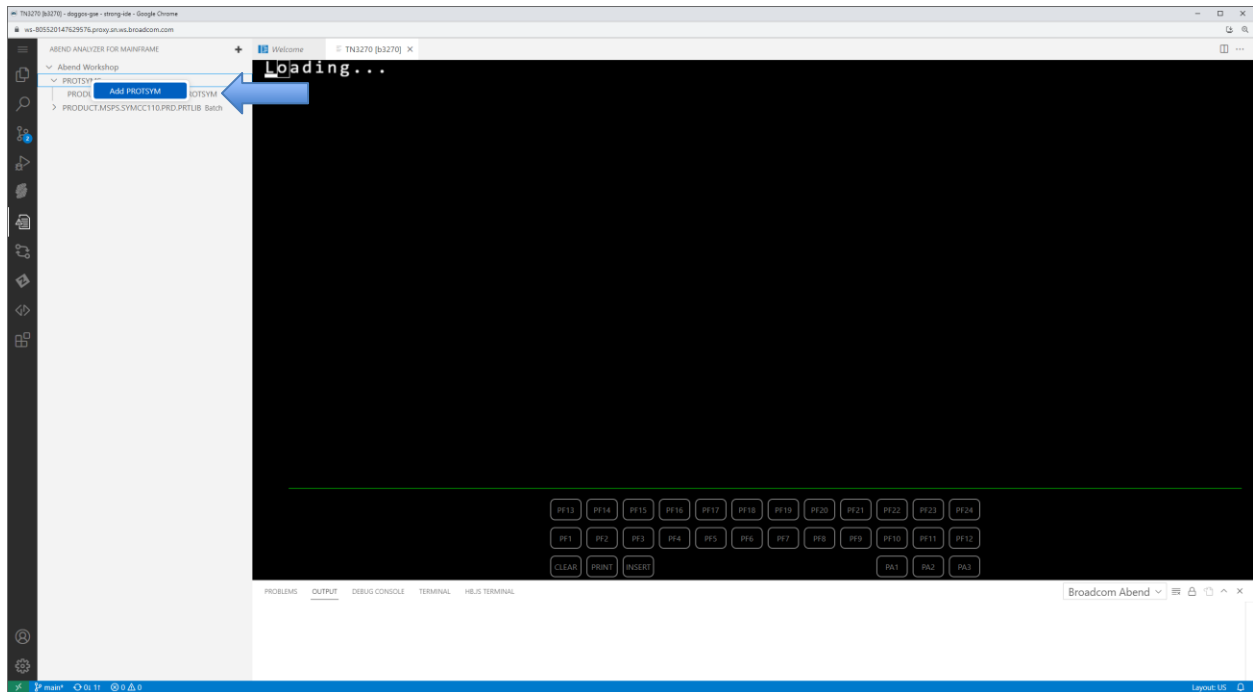
Then Abend Analyzer needs to know if these dumps are batch or CICS. For this workshop batch dumps were captured.



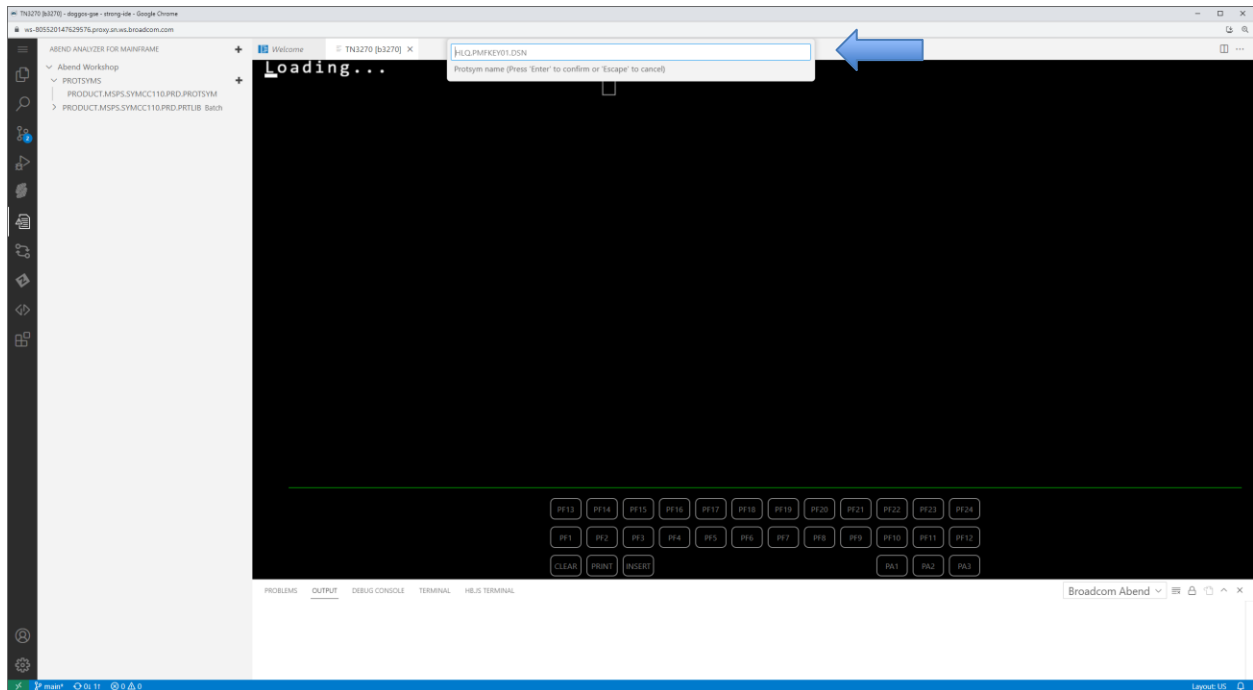
That will return with the prtlib added to Abend Analyzer.



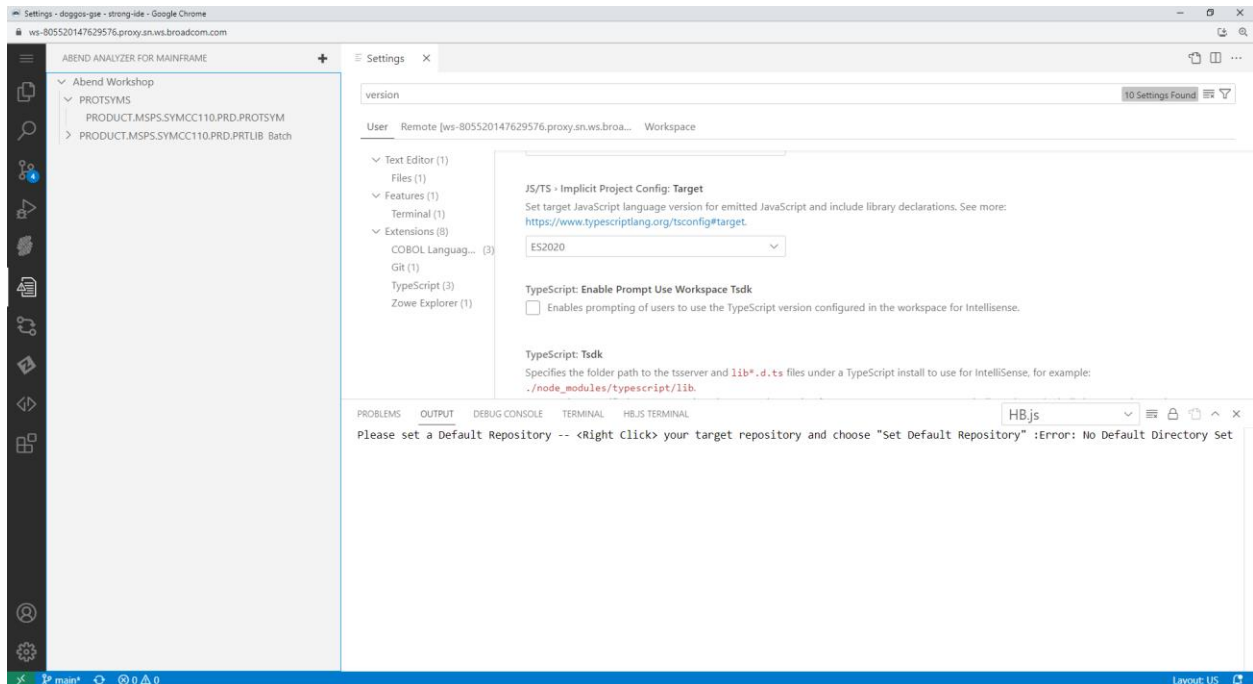
The next file that needs to be added is the PROTSYM. This file is where the compiled version of the COBOL program is stored. To add the PROTSYM file right click the PROTSYMS on the left. Then click the Add PROTSYM button.



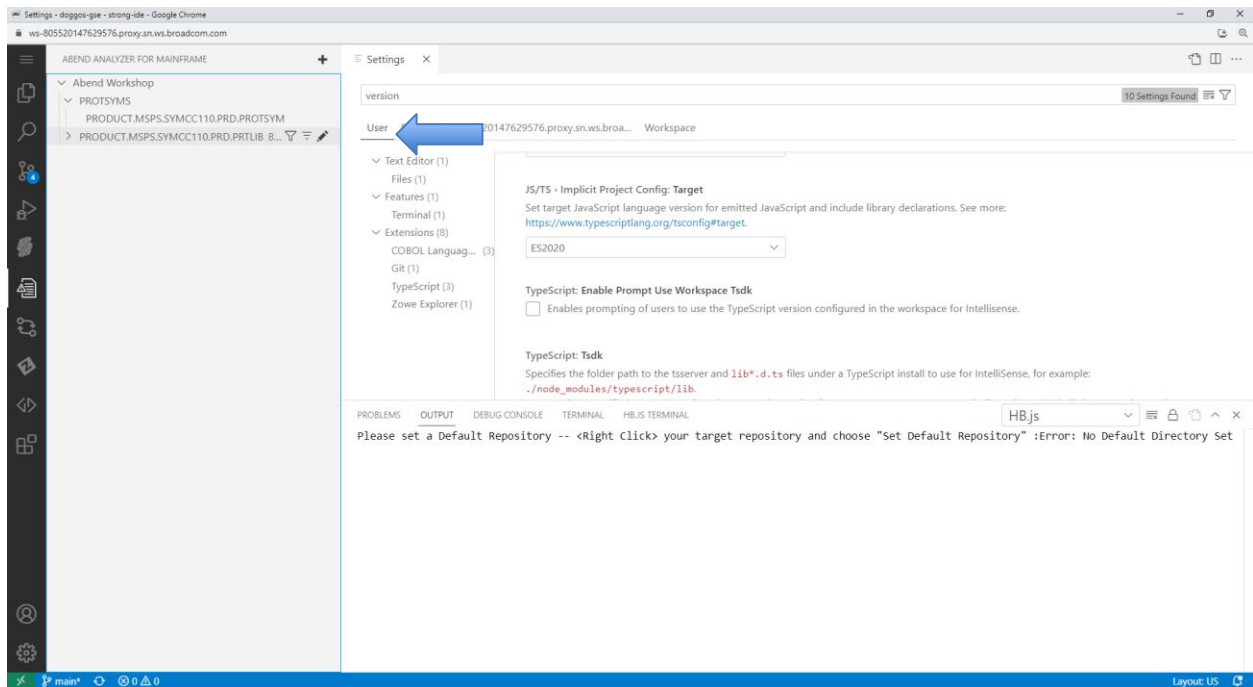
The following popup will appear on screen. Add the PROTSYMS dataset
PRODUCT.MSPS.SYMCC110.PRD.PROTSYM



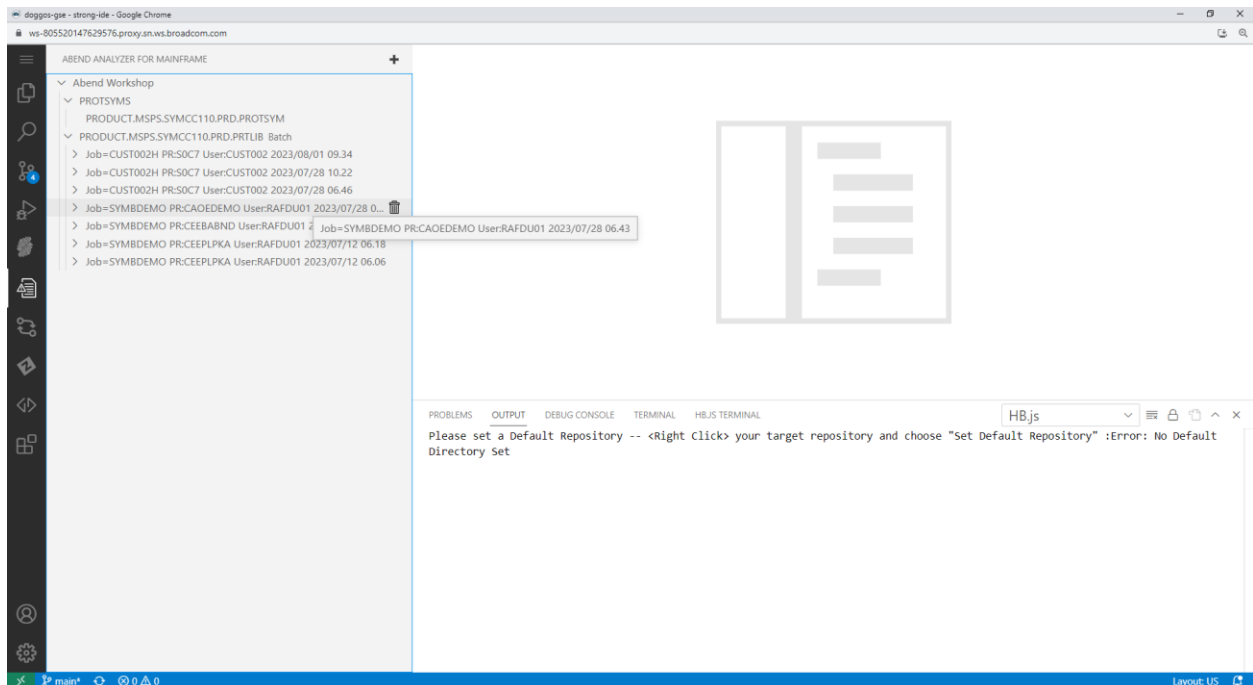
After pressing enter you will see the PRTLIB and PROTSYM files listed.



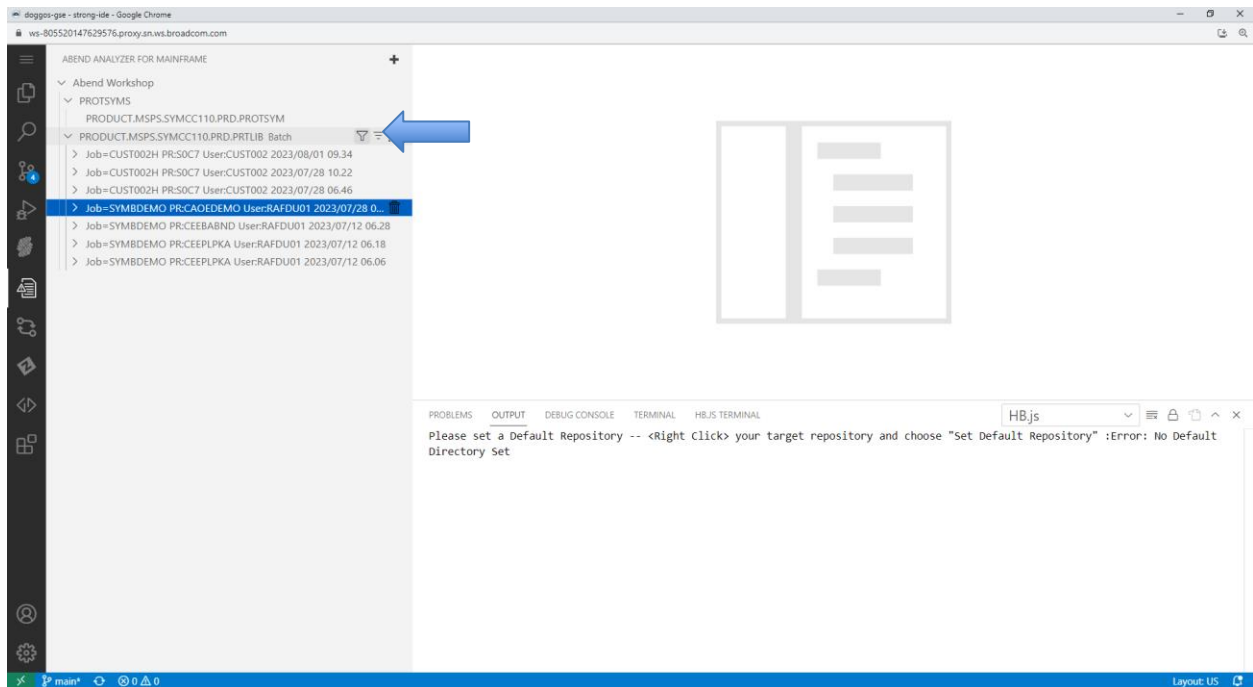
To access the dumps in the PRTLIB file expand the dataset.



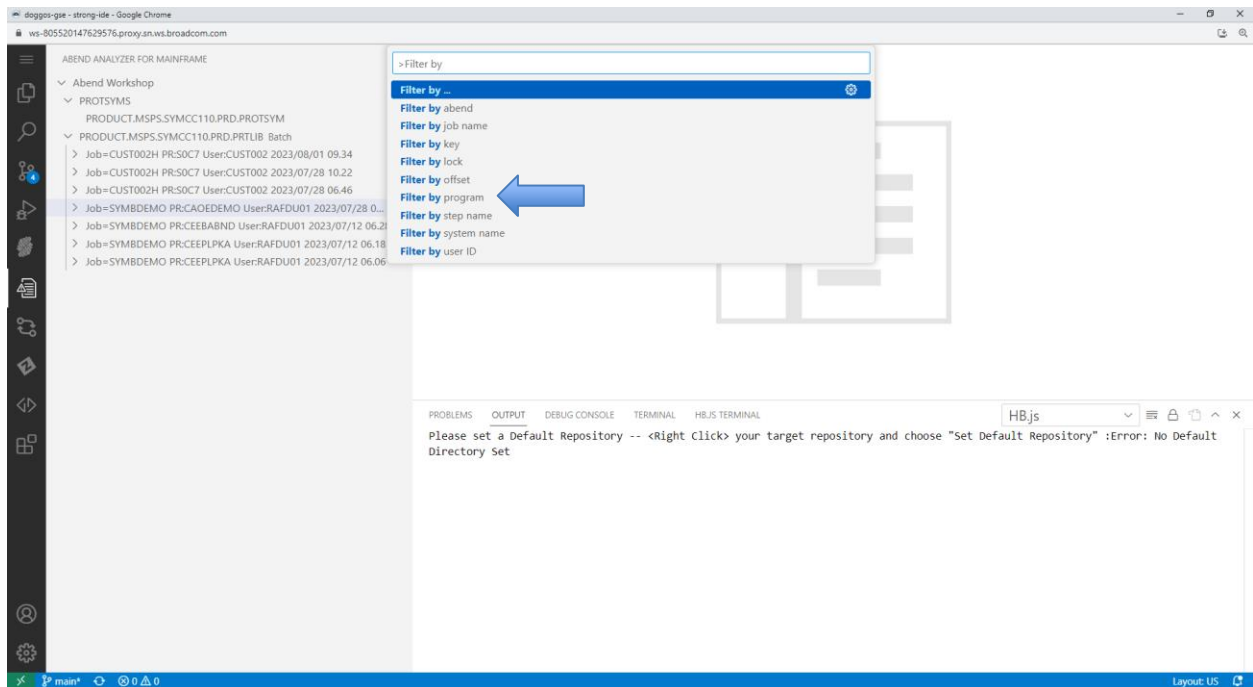
This will sign you on to the mainframe and access the dump file.



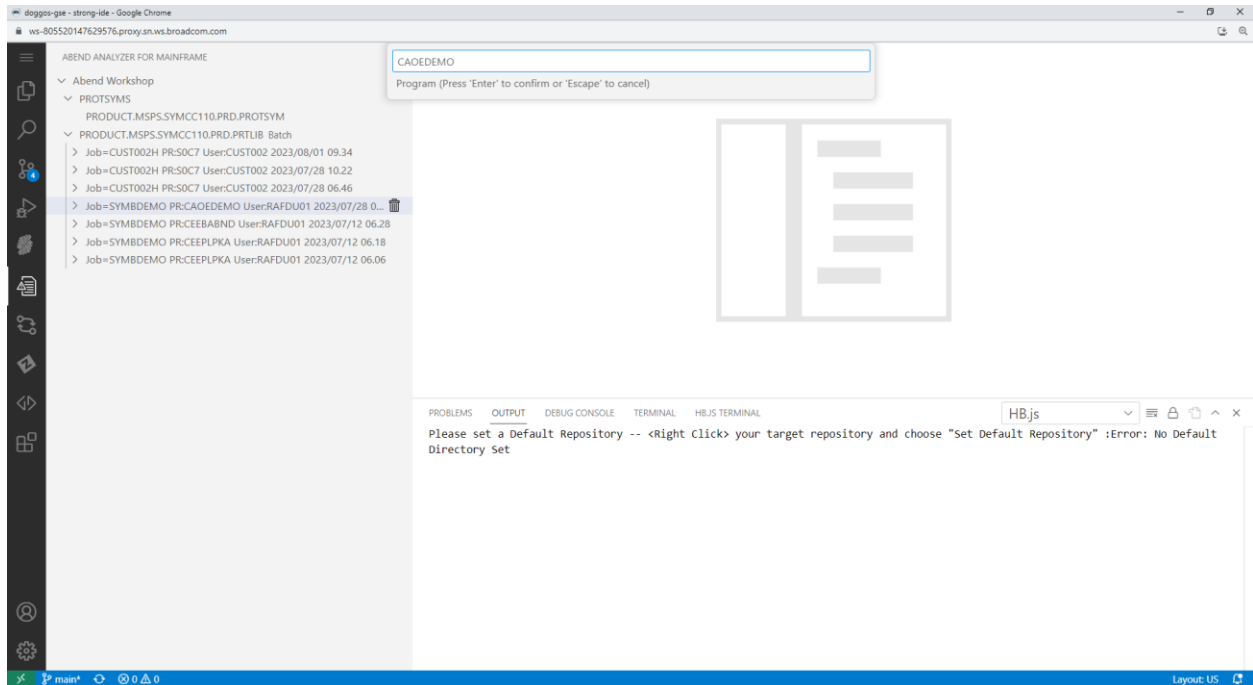
Depending on the system this file can be large. At this point you can filter the dump reports by clicking on the Filter by button.



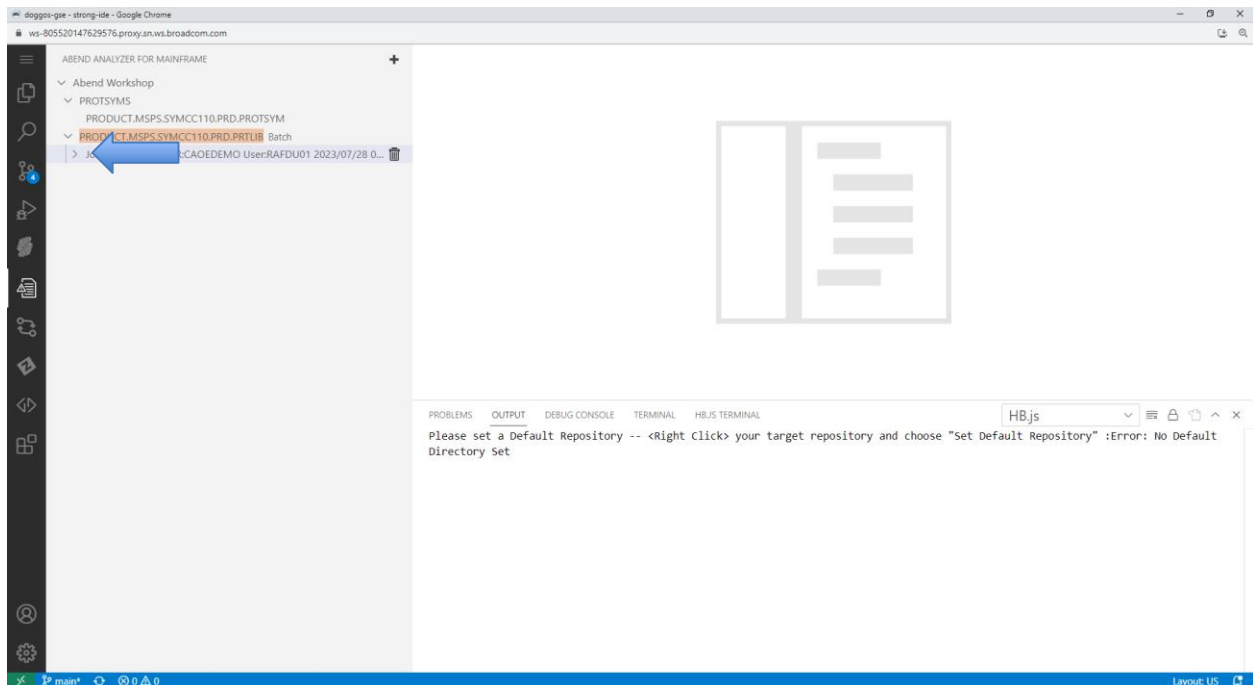
At this point you can filter by different fields. For this exercise click on Filter by program



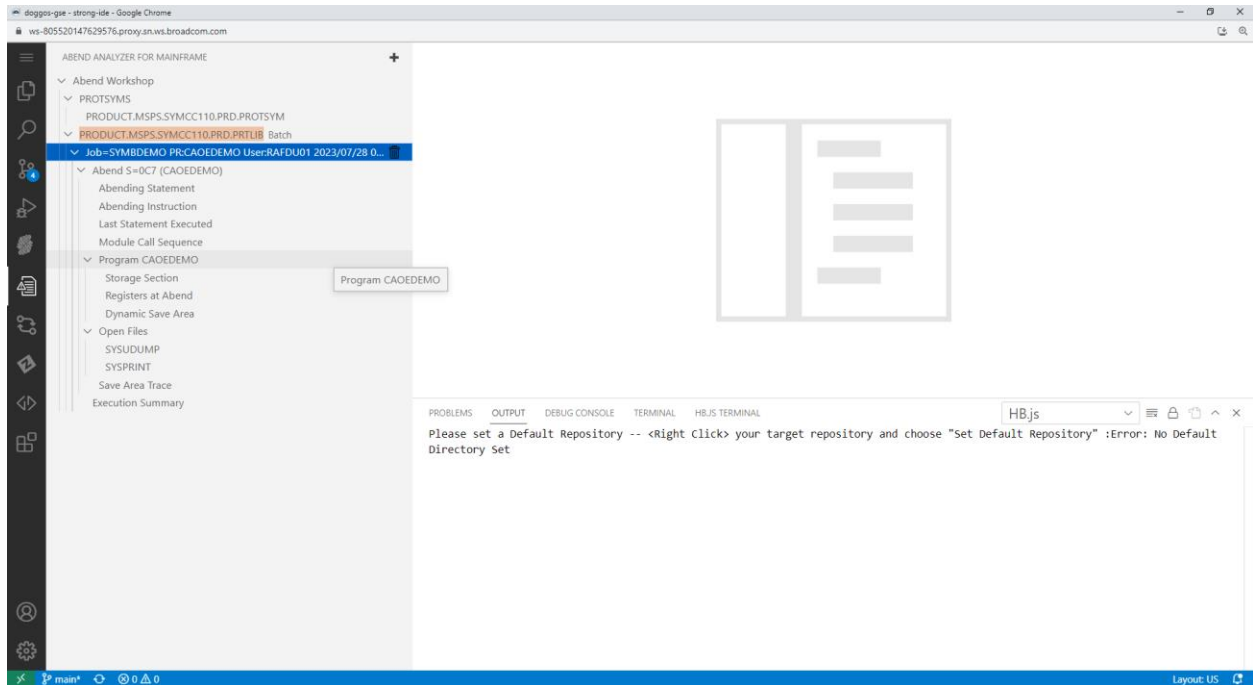
Enter CAOEDEMO in the pop up box.



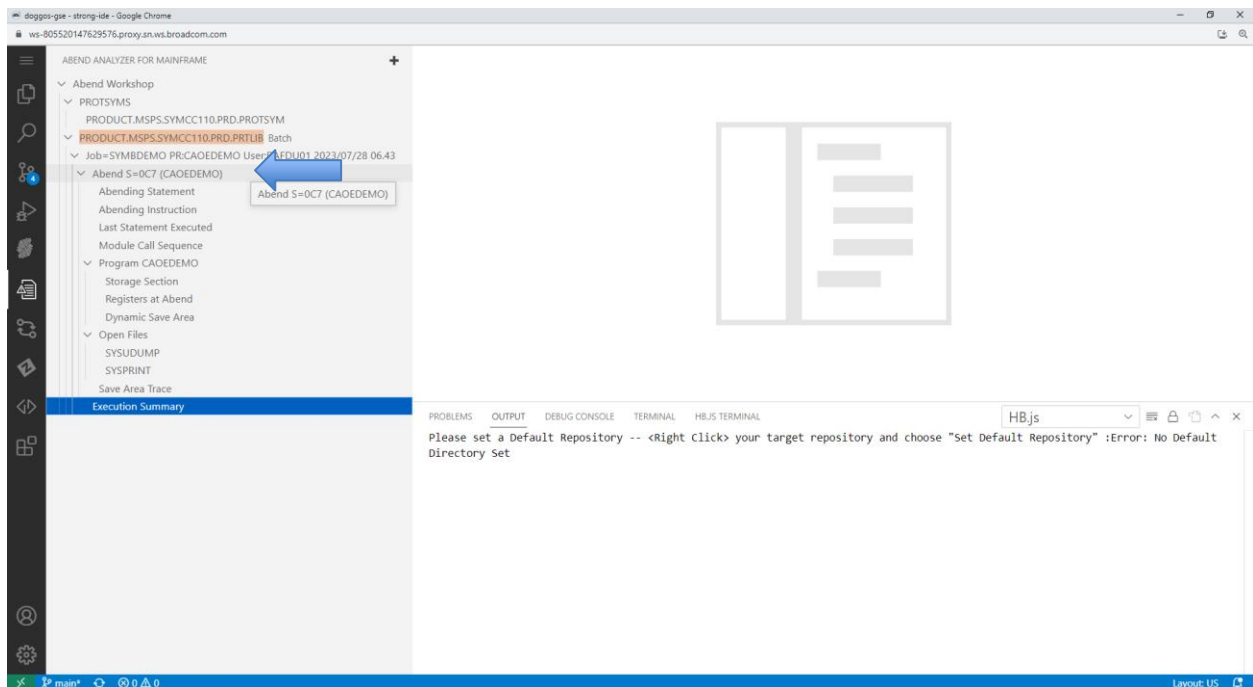
This reduced the number of dump reports and narrows the list to the dump report that you are looking for.



This will put the dump report into a tree for easier navigation.



Click on the first time in the tree Abend S=0C7 (CAOEDEMO).



This will bring up a report showing the following information.

1. Abend code and program name.
2. Description of the abend.

3. Possible causes for the abend.
4. What to do to correct the issue.

The screenshot displays the 'ABEND ANALYZER FOR MAINFRAME' application. The left sidebar shows a tree view with the following structure:

- Abend Workshop
 - PROTSYMS
 - PRODUCT.MSPS.SYMCCT110.PRO.PRTLIB
 - PRODUCT.MSPS.SYMCCT110.PRO.PRTLIB Batch
 - Job=SYMBDEMO PR:CAOEDEMO User:RAFDU01 2023/07/28 06:43
 - Abend S=0C7 (CAOEDEMO)
 - Abending Statement
 - Abending Instruction
 - Last Statement Executed
 - Module Call Sequence
 - Program CAOEDMO
 - Storage Section
 - Registers at Abend
 - Dynamic Save Area
 - Open Files
 - SYSUDUMP
 - SYSPRINT
 - Save Area Trace
 - Execution Summary

The main window shows the analysis of the S0C7 abend. The top section displays the program name 'CAOEDEMO', offset '0003B76', and time '06:43'. Below this, the abend code 'S-0C7' is shown. The 'DESCRIPTION' section states: 'S0C7 - DATA EXCEPTION: A NUMERIC FIELD CONTAINED NON-NUMERIC DATA.' The 'POSSIBLE CAUSES' section lists seven items:

1. NUMERIC DATA WAS NOT INITIALIZED, OR VARIABLE DATA NOT NUMERIC.
2. ATTEMPTED TO PERFORM A DECIMAL ARITHMETIC INSTRUCTION ON EITHER AN UNSIGNED OR UNPACKED FIELD.
3. A SUBSCRIPT OR INDEX CONTAINED AN INVALID VALUE:
 - A. FAILURE TO INITIALIZE A SUBSCRIPT.
 - B. INITIALIZED A SUBSCRIPT TO ZERO, BUT FAILED TO ADD 1 BEFORE ITS FIRST USE.
 - C. NOT REINITIALIZING A SUBSCRIPT AFTER A LOOP USING IT.
 - D. SUBSCRIPTING BY INPUT DATA THAT WAS NOT CHECKED FIRST FOR A VALID RANGE OF VALUES.
 - E. COMPUTING THE SUBSCRIPT WITHOUT CHECKING THE RESULTS FOR A VALID RANGE OF VALUES.
4. A COMP-3 FIELD HAD AN INVALID SIGN.
5. A GROUP LEVEL MOVE FROM DISPLAY FIELD TO COMP OR COMP-3 FIELD THEREFORE NO CONVERSION WAS PROVIDED.
6. OMISSION OF A USAGE CLAUSE WIPING OUT WORKING STORAGE BY REFERENCING A RECORD NAME BEFORE ITS FILE IS OPENED.
7. WIPING OUT LITERAL POOL BY SUBSCRIPTING BEYOND DATA AREAS.

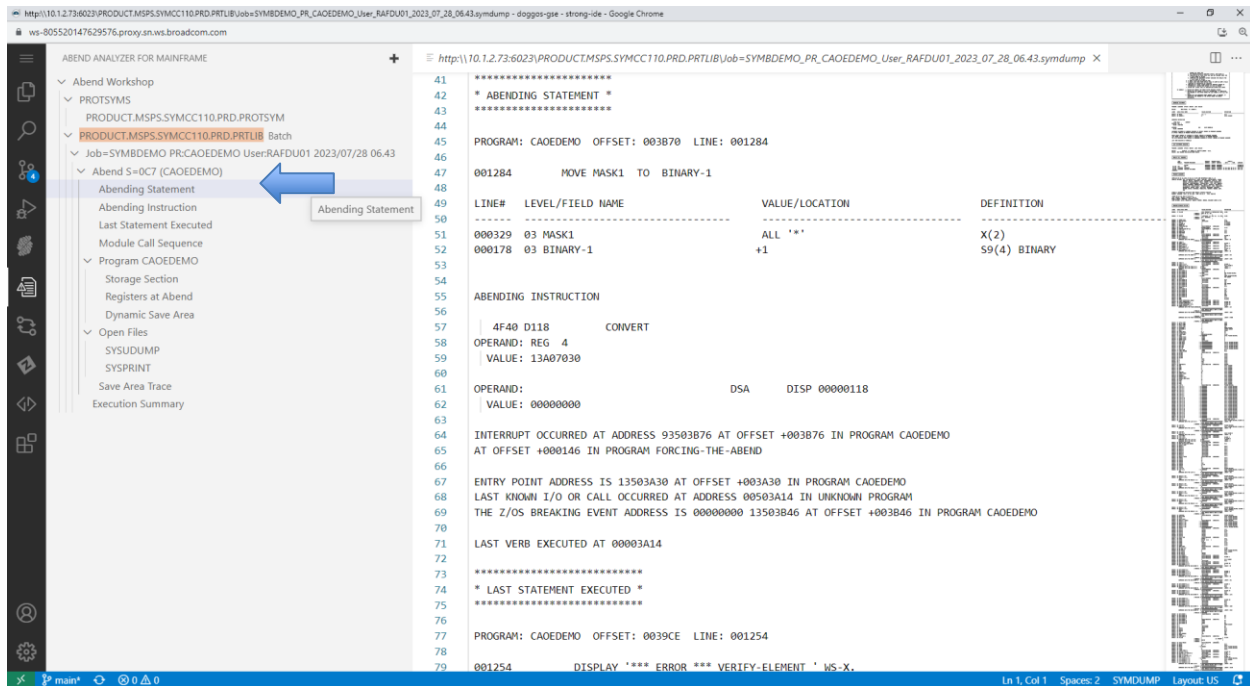
The 'TO CORRECT' section lists three items:

1. REVIEW AND CORRECT THE ABOVE STATED POSSIBLE CAUSES.
2. REVIEW THE LAST KNOWN I/O OR CALL INFORMATION. IT IDENTIFIES THE PROGRAM AT THE OFFSET WHERE THE LAST KNOWN I/O OR CALL WAS MADE.
3. REVIEW THE Z/OS BREAKING EVENT ADDRESS LINE. IF PRESENT, IT IDENTIFIES THE PROGRAM AT THE OFFSET WHERE THE BRANCH ORIGINATED.

To close this section of the report click the x at the top of the report tab.

This screenshot is identical to the one above, but with a blue arrow pointing to the 'X' button in the top right corner of the report tab, indicating how to close the report.

Now that you know you have 0C7 which is a data exception. Let see where in the program it is. From the dump tree click on Abending Statement.



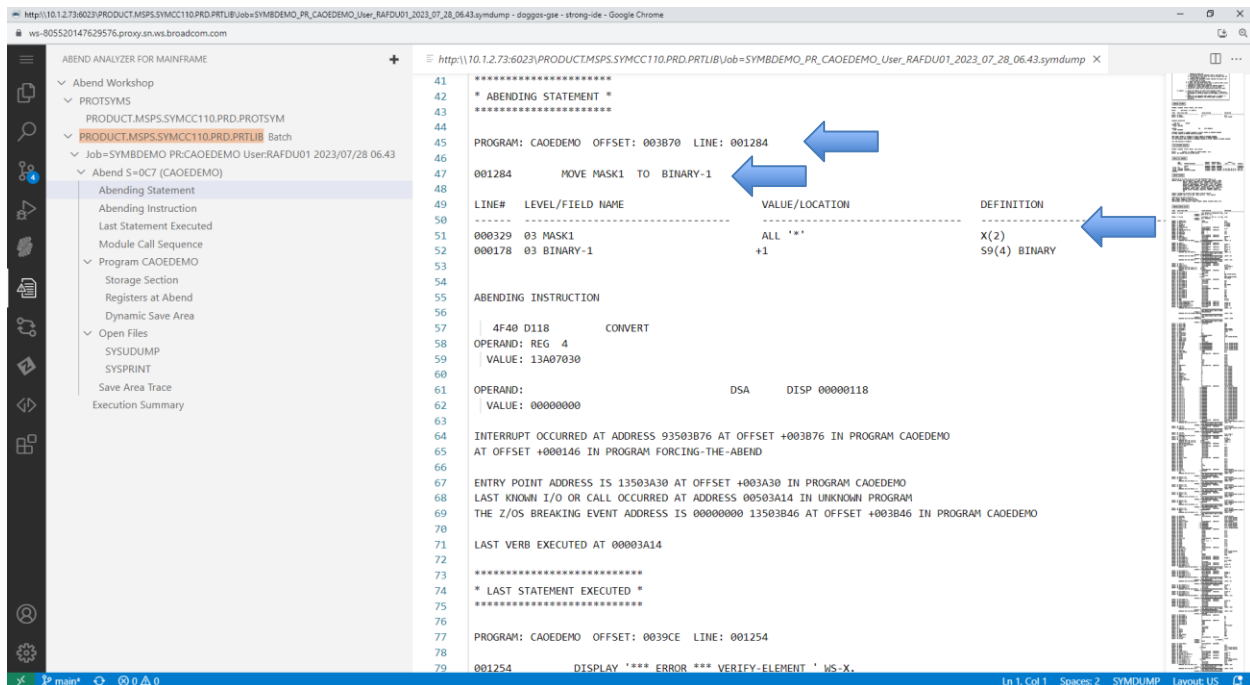
The screenshot shows the ABEND ANALYZER FOR MAINFRAME interface. On the left, a tree view shows the hierarchy: Abend Workshop > PROTSYMS > PRODUCT.MSPS.SYMCC110.PROD.PRTLIB > Job=SYMBDEMO PR:CAOEDEMO User:RAFDU01 2023/07/28 06:43 > Abend S=0C7 (CAOEDEMO) > Abending Statement. A blue arrow points to the 'Abending Statement' item. The main pane displays the following text:

```
41 *****
42 * ABENDING STATEMENT *
43 *****
44
45 PROGRAM: CAOEDMO  OFFSET: 003B70  LINE: 001284
46
47 001284      MOVE MASK1  TO BINARY-1
48
49
50 LINE#  LEVEL/FIELD NAME                                VALUE/LOCATION                                DEFINITION
51 -----
52 000329  03 MASK1                                         ALL '*'                                       X(2)
53 000178  03 BINARY-1                                     +1                                           S9(4) BINARY
54
55 ABENDING INSTRUCTION
56
57 4F40 D118      CONVERT
58 OPERAND: REG 4
59 VALUE: 13A07030
60
61 OPERAND:                                     DSA      DISP 00000118
62 VALUE: 00000000
63
64 INTERRUPT OCCURRED AT ADDRESS 93503B76 AT OFFSET +003B76 IN PROGRAM CAOEDMO
65 AT OFFSET +000146 IN PROGRAM FORCING-THE-ABEND
66
67 ENTRY POINT ADDRESS IS 13503A30 AT OFFSET +003A30 IN PROGRAM CAOEDMO
68 LAST KNOWN I/O OR CALL OCCURRED AT ADDRESS 00503A14 IN UNKNOWN PROGRAM
69 THE Z/OS BREAKING EVENT ADDRESS IS 00000000 13503B46 AT OFFSET +003B46 IN PROGRAM CAOEDMO
70
71 LAST VERB EXECUTED AT 00003A14
72
73 *****
74 * LAST STATEMENT EXECUTED *
75 *****
76
77 PROGRAM: CAOEDMO  OFFSET: 0039CE  LINE: 001254
78
79 001254      DISPLAY ***** ERROR ***** VERIFY-ELEMENT 'HS-X,
```

As you can see it is showing you the program CAOEDMO the offset 003B70 and the line 001284. Plus the actual line of code.

001284 MOVE MASK1 TO BINARY-1

Then just below that it is show the data fields with line, level/field name, value/location and definition for this statement.



ABEND ANALYZER FOR MAINFRAME

- Abend Workshop
- PROTSYMS
 - PRODUCT.MSPS.SYMC110.PROD.PROTSYM
 - PRODUCT.MSPS.SYMC110.PROD.PRTLIB Batch
 - Job=SYMBDEMO PR:CAOEDEMO User:RAFDU01 2023/07/28 06.43
 - Abend S=0C7 (CAOEDEMO)
 - Abending Statement
 - Abending Instruction
 - Last Statement Executed
 - Module Call Sequence
 - Program CAOEDMO
 - Storage Section
 - Registers at Abend
 - Dynamic Save Area
 - Open Files
 - SYSUDUMP
 - SYSPRINT
 - Save Area Trace
 - Execution Summary

Job=SYMBDEMO PR:CAOEDEMO User:RAFDU01 2023/07/28 06.43

Abend S=0C7 (CAOEDEMO)

Abending Statement

Abending Instruction

Last Statement Executed

Module Call Sequence

Program CAOEDMO

Storage Section

Registers at Abend

Dynamic Save Area

Open Files

SYSUDUMP

SYSPRINT

Save Area Trace

Execution Summary

PROGRAM: CAOEDMO OFFSET: 003B70 LINE: 001284

001284 MOVE MASK1 TO BINARY-1

LINE# LEVEL/FIELD NAME VALUE/LOCATION DEFINITION

000329 03 MASK1 ALL '*' X(2)

000178 03 BINARY-1 +1 59(4) BINARY

ABENDING INSTRUCTION

4F40 D118 CONVERT

OPERAND: REG 4

VALUE: 13A07030

OPERAND: DSA DISP 00000118

VALUE: 00000000

INTERRUPT OCCURRED AT ADDRESS 93503B76 AT OFFSET +003B76 IN PROGRAM CAOEDMO

AT OFFSET +000146 IN PROGRAM FORCING-THE-ABEND

ENTRY POINT ADDRESS IS 13503A30 AT OFFSET +003A30 IN PROGRAM CAOEDMO

LAST KNOWN I/O OR CALL OCCURRED AT ADDRESS 00503A14 IN UNKNOWN PROGRAM

THE Z/OS BREAKING EVENT ADDRESS IS 00000000 13503B46 AT OFFSET +003B46 IN PROGRAM CAOEDMO

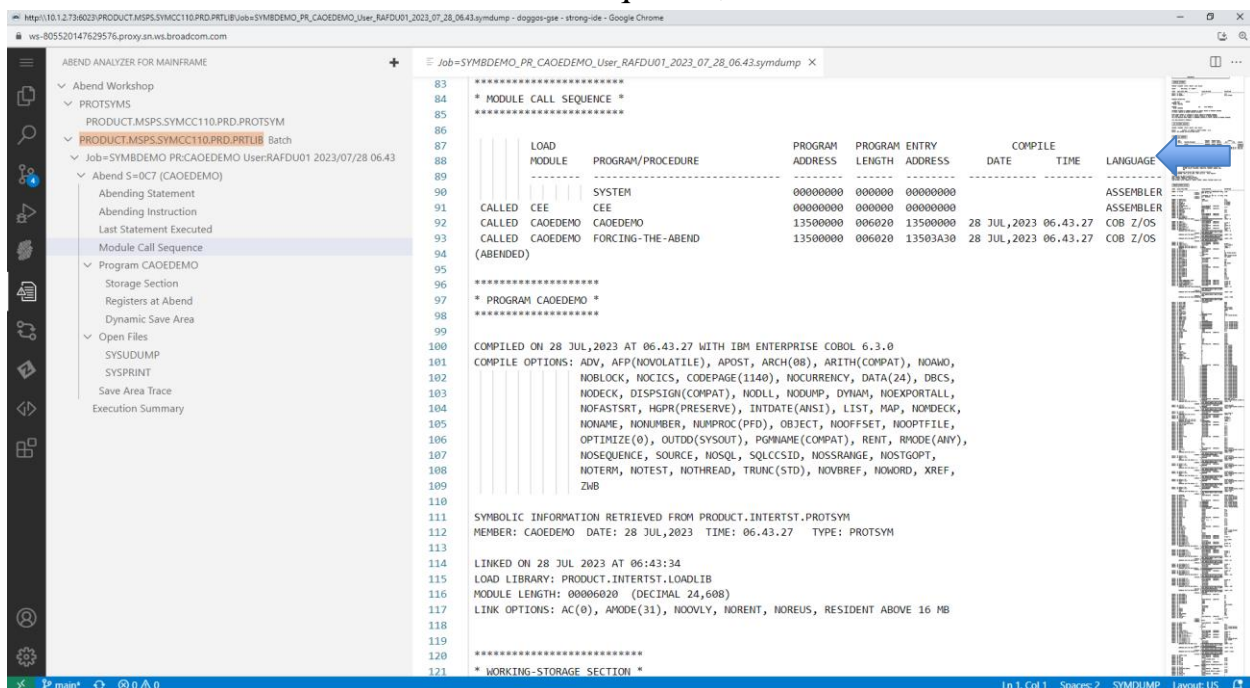
LAST VERB EXECUTED AT 00003A14

* LAST STATEMENT EXECUTED *

PROGRAM: CAOEDMO OFFSET: 0039CE LINE: 001254

001254 DISPLAY '*** ERROR *** VERIFY-ELEMENT ' WS-X,

Close the report tab like before. The next item that provides very useful information is the Module Call Sequence, click on it.



ABEND ANALYZER FOR MAINFRAME

- Abend Workshop
- PROTSYMS
 - PRODUCT.MSPS.SYMC110.PROD.PROTSYM
 - PRODUCT.MSPS.SYMC110.PROD.PRTLIB Batch
 - Job=SYMBDEMO PR:CAOEDEMO User:RAFDU01 2023/07/28 06.43
 - Abend S=0C7 (CAOEDEMO)
 - Abending Statement
 - Abending Instruction
 - Last Statement Executed
 - Module Call Sequence
 - Program CAOEDMO
 - Storage Section
 - Registers at Abend
 - Dynamic Save Area
 - Open Files
 - SYSUDUMP
 - SYSPRINT
 - Save Area Trace
 - Execution Summary

Job=SYMBDEMO PR:CAOEDEMO User:RAFDU01 2023/07/28 06.43

Abend S=0C7 (CAOEDEMO)

Abending Statement

Abending Instruction

Last Statement Executed

Module Call Sequence

Program CAOEDMO

Storage Section

Registers at Abend

Dynamic Save Area

Open Files

SYSUDUMP

SYSPRINT

Save Area Trace

Execution Summary

* MODULE CALL SEQUENCE *

LOAD	MODULE	PROGRAM/PROCEDURE	PROGRAM ADDRESS	PROGRAM LENGTH	ENTRY ADDRESS	DATE	TIME	LANGUAGE
	CEE	CEE	00000000	000000	00000000			ASSEMBLER
	CAOEDEMO	CAOEDEMO	00000000	000000	00000000			ASSEMBLER
	CAOEDEMO	FORCING-THE-ABEND	13500000	006020	13500000	28 JUL,2023	06.43.27	COB Z/OS
	(ABENDED)		13500000	006020	13503A30	28 JUL,2023	06.43.27	COB Z/OS

* PROGRAM CAOEDMO *

COMPILED ON 28 JUL,2023 AT 06.43.27 WITH IBM ENTERPRISE COBOL 6.3.0

COMPILE OPTIONS: ADV, AFP(NOVOLATILE), APOST, ARCH(08), ARITH(COMPAT), NOAWO, NOBLOCK, NOCICS, CODEPAGE(1140), NOCURRENCY, DATA(24), DBCS, NODECK, DISP(SIGN(COMPAT)), MODLL, NODUMP, DYNAM, NOEXPORTALL, NOFASTSR, HGR(PRESERVE), INTDATE(ANIS), LIST, MAP, NOHDECK, NOHNAME, NOHNUMBER, NUPROG(PFD), OBJECT, NOOFFSET, NOOPTFILE, OPTIMIZE(0), OUTDO(SVSOUT), PGMNAME(COMPAT), REIT, RMODE(ANY), NOSEQUENCE, SOURCE, NOSQL, SQLCCSID, NOSSRANGE, NOSTGOPT, NOTERM, NOTEST, NOTHREAD, TRUNC(STD), NOVBREF, NOWORD, XREF, ZWB

SYMBOLIC INFORMATION RETRIEVED FROM PRODUCT.INTERST.PROTSYM

MEMBER: CAOEDMO DATE: 28 JUL,2023 TIME: 06.43.27 TYPE: PROTSYM

LINKED ON 28 JUL 2023 AT 06:43:34

LOAD LIBRARY: PRODUCT.INTERST.LOADLIB

MODULE LENGTH: 00006020 (DECIMAL 24,608)

LINK OPTIONS: AC(0), AMODE(31), NOOVLY, NORENT, NOREUS, RESIDENT ABOVE 16 MB

* WORKING-STORAGE SECTION *

In this part of the report it tells you program, compile information and language. Close the report.

Congratulations you have completed the basic workshop.