Lab 1. Setting up VSCode and Extensions

Overview

This lab covers all aspects of the download and installation of Visual Studio (VS) Code and the necessary prerequisites. This lab will cover the necessary steps and information to download and install the prerequisites needed for the subsequent labs within this course. This software is needed for one or more of the applications we will be utilizing in our labs throughout the course.

Objectives

- Install node.js
- Install Java SDK
- Install and run Visual Studio Code
- Install Zowe Explorer plugin
- Install IBM Z Open Editor plugin
- Complete registration

Lab instructions

Install node.js:

1. Check for node.js installation and verify that the version number is v8 or higher.

Open your workstation's version of the command prompt (called Terminal on Mac OS X). Once the command prompt is open, use the command in Example 1 to check if your workstation currently has a version of node.js installed.

Example 1. Node.js version

```
C:\Users\User> node -v

V12.16.1
```

If you do not see a version number after you submit the command, you do not have node.js installed, or if it shows a version less than v8, you should continue following these instructions. If you do see a version number and it is v8 or higher, you can move on to section Install Java SDK.

2. If node.js version is less than v8, or node isn't installed at all.

Updating node.js to the appropriate version number is a relatively simple process because the installer takes care of most of the "heavy lifting". All you will need to do is visit the Node.js download site, provided below, and follow the download and installation instructions for your specific workstation platform. Do this same process if you do not already have node.js installed.

https://nodejs.org/en/download/

This process will install the latest versions of Node.js and the node package manager (npm) and overwrite any older version files in your system. This removes the step of needing to manually uninstall the previous versions beforehand.

3. Once completed, verify the installation and proper version number, as shown previously in Example 1.

Note: The version numbers in our examples are provided purely for reference and may not reflect the latest versions of the software.

Install Java SDK:

1. Check for Java installation and verify that the version number is v8 or higher.

Open your workstation's version of the command prompt, if not already open. Once the command prompt is open, use the command in Example 2 to check if your workstation currently has a version of Java installed. Java SDK 8 is the preferred version for these labs, however, any versions higher than that will suffice.

Example 1. Java version

```
C:\Users\User> java -version

java version "1.8.0_241"

Java(TM) SE Runtime Environment (build 1.8.0_241-b07)

Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)
```

If you do not see a version number after you submit the command, you do not have Java installed or if it shows a version less than v8, you should continue following these instructions. The display format of the version number for Java is slightly different than what is displayed for node.js. With Java, the second value in the displayed version number, i.e. the "8" in Example 2, is the version number. So, our example is showing Java SDK version 8. If you do see a version number and it is v8 or higher, you can move on to section Install VSCode.

2. If your version of Java displayed is less than v8, you need to uninstall the current version on your workstation and reinstall the correct version. Follow the link below to uninstall instructions that represent your workstation operating system (OS).

   Linux: https://www.java.com/en/download/help/linux_uninstall.xml
   Mac: https://www.java.com/en/download/help/mac_uninstall_java.xml
   Windows: https://www.java.com/en/download/help/uninstall_java.xml

3. Once Java is uninstalled from your workstation, you can click the Java JDK 8 download link below and follow the installation instructions for your specific OS.

   https://www.oracle.com/java/technologies/javase-jdk8-downloads.html

4. Verify the installation and proper version number as shown in Example 2.

Note: You will be prompted to register a new Oracle account in order to download the installation file, please do so. If you have an existing account, you may use that to log in and continue.

Install VSCode

If you do not already have VSCode installed on your workstation, please do so now by following the download and installation instructions at the link below:

https://code.visualstudio.com/download

Figure 1. *VSCode download site*

Note: Be sure to select the correct installation file for your workstations respective OS, shown in **Error! Reference source not found.**

Install VSCode extensions

This section introduces two VSCode extensions, Zowe Explorer and IBM Z Open Editor, listed in Figure 2, and instructions on how to install them.

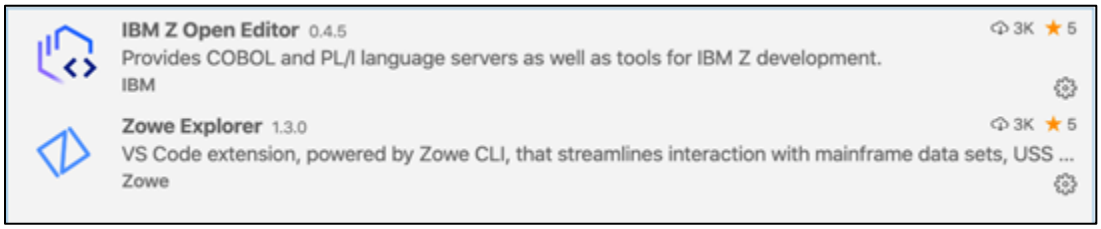

Figure 2. *VSCode required extensions*

Zowe Explorer:

Zowe is a new, and the first open source framework for z/OS and provides solutions for development and operations teams to securely manage, control, script and develop on the mainframe like any other cloud platform. Out of the box, the Zowe Explorer provides a lot of functionality allowing z/OS developers access to jobs, datasets and (USS) files on a z/OS server. Backed by the Zowe CLI and z/OSMF, developers now have powerful features that makes it easy to work with z/OS within the familiar VSCode environment. This extension can be used to edit COBOL and PL/I files opened on z/OS MVS™ and USS using the Zowe extension's Data

Sets and USS views. It can even run JCL and lets you browse job spool files. For more information on Zowe Explorer and its interaction with z/OS please visit:

https://ibm.github.io/zopeneditor-about/Docs/zowe_interactwithzos.html

Install Zowe Explorer:

Open VSCode and in the left side tool menu select **Extensions**. From there, in the "Search Extensions in Marketplace" search field, type Zowe Explorer. Search results will begin populating, select **"Zowe Explorer"** and click **install**, depicted in Figure 3.

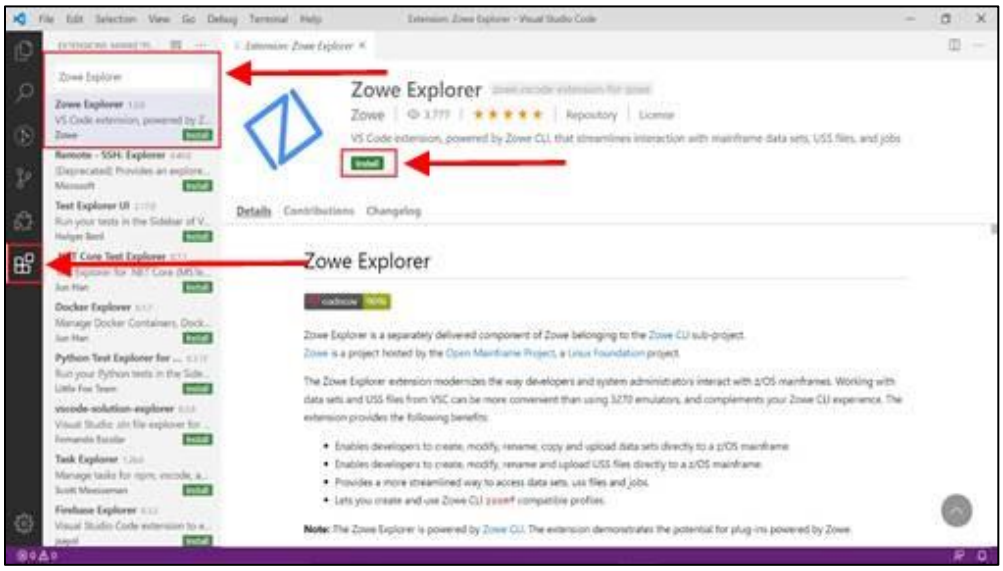

Figure 3. *Install Zowe Explorer in VSCode*

The Zowe community has a number of on-line videos that walk through the steps required to install, configure and operate the Zowe Explorer, see:
http://www.youtube.com/watch?v=G_WCsFZIWt4&t=0m38s

IBM Z Open Editor:

IBM Z Open Editor brings COBOL and PL/I language support to Microsoft VSCode. It is one of the several next generation capabilities for an open development experience for z/OS®. It also works in association with the Zowe Explorer plugin. For more information on IBM Z Open Editor, please visit:

https://ibm.github.io/zopeneditor-about/Docs/introduction.html#key-capabilities

Install IBM Z Open Editor:

Open VSCode and in the left side tool menu select **Extensions**. From there, in the "Search Extensions in Marketplace" search field, type IBM Z Open Editor. Search results will begin populating, select **" IBM Z Open Editor "** and click **install**, depicted in Figure 4.

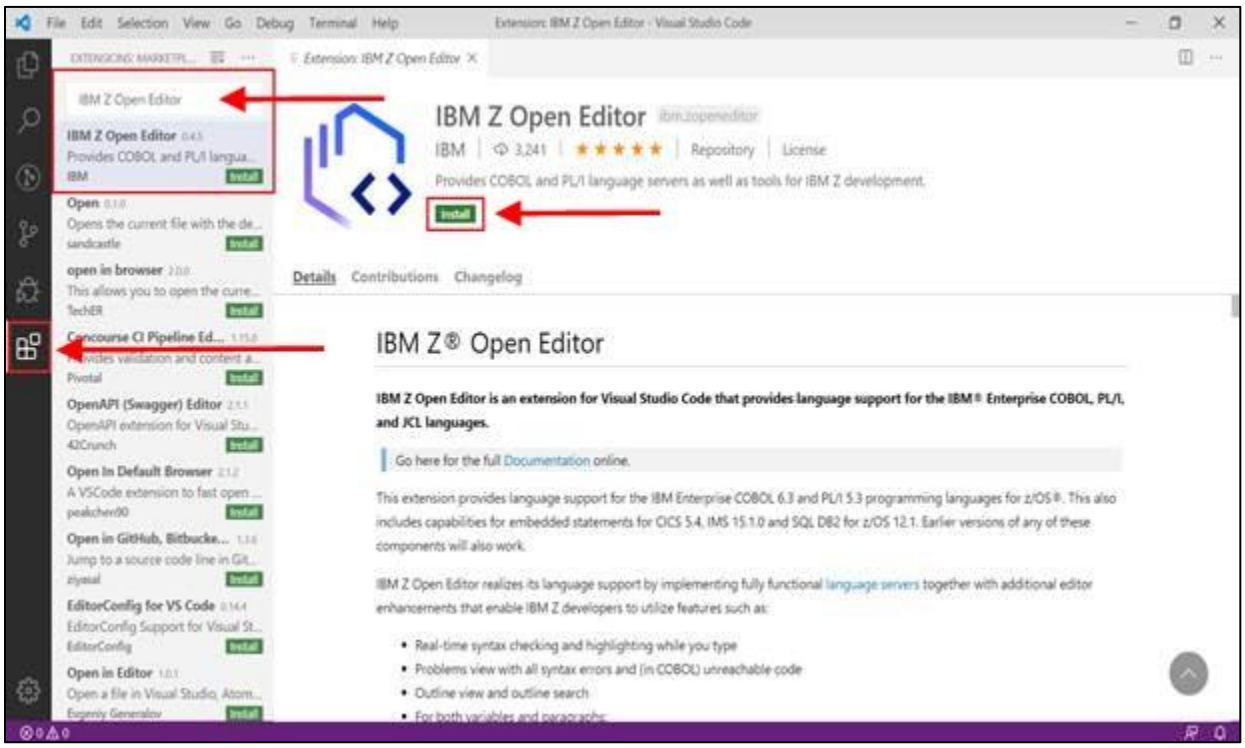

Figure 4. *Install IBM Z Open Editor in VSCode*

Note: There may be some limitations with IBM Z Open Editor if running a 32-bit Java version on Windows.

Register for an account:

Follow the link provided below to register for an account with the mainframe. This account registration will provide you with a username, password and z/OSMF URL to establish your connection to the mainframe and must be done before moving on to lab 2.

Registration Link:
https://www-01.ibm.com/events/wwe/ast/mtm/cobolvscode.nsf/enrollall?openform

Lab 2. Connecting to the Mainframe

Overview

In this lab exercise you will connect to an IBM Z mainframe system, view a simple COBOL hello world program in VSCode, submit JCL to compile the COBOL program, and view the output.

Objectives

- Setup connection profile in Zowe Explorer
- Connect to z/OS through connection profile
- Filter data sets
- Submit “hello world” job
- View jobs output

Lab instructions

1. The lab assumes installation of VSCode with Z Open Editor and Zowe Explorer extensions as shown in **Error! Reference source not found.**

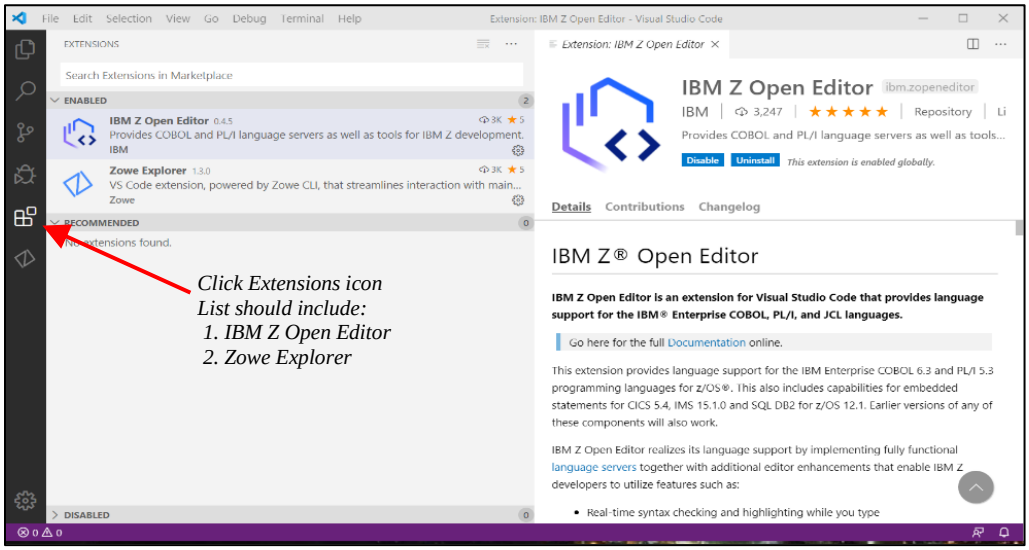

Figure 1. *VSCode extensions*

2. Click the Zowe Explorer icon as shown in Figure 2.

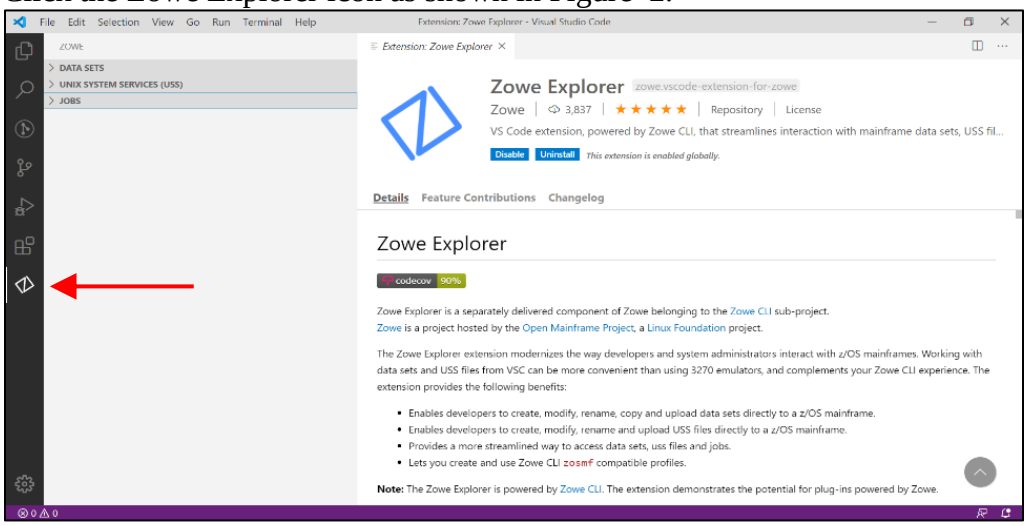

Figure 2. *Zowe Explorer icon*

3. Zowe Explorer can list Data Sets, Unix System Services (USS) files, and Jobs output as shown in Figure 3. a "+" will appear when hovering to the far right of the DATA SETS line. Click the + to define a VSCode profile.

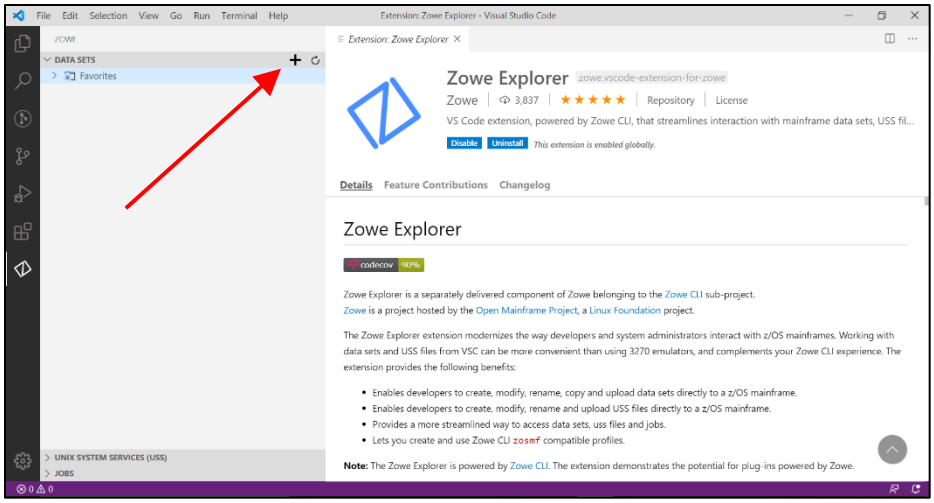

Figure 3. *Zowe Explorer*

4. A box appears to define a new profile. Click + to the left of Create a New Connection to z/OS as shown in Figure 4.

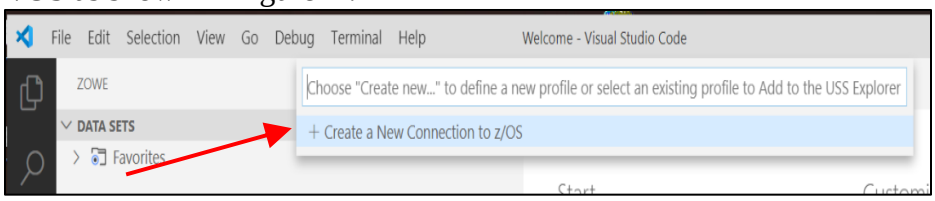

Figure 4. *Create a new connection to z/OS*

5. Select a name to enter, then press enter. Figure 5. uses LearnCOBOL as the selected connection name.

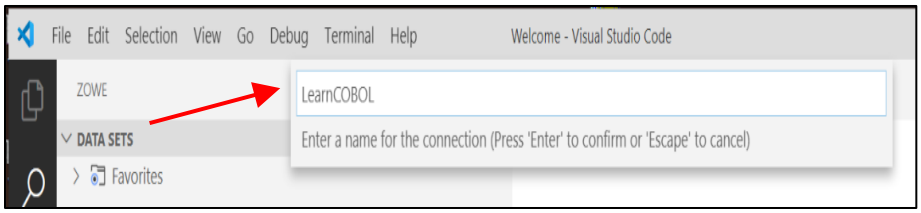

Figure 5. *Set connection name*

6. VSCode prompts for a z/OSMF URL and port as shown in Figure 6. The z/OSMF URL and port will be provided in your account registration confirmation email.

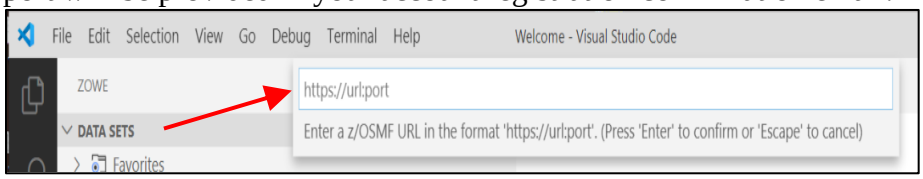

Figure 6. *z/OSMF URL*

7. A sample z/OSMF URL and port is entered in Figure 7.

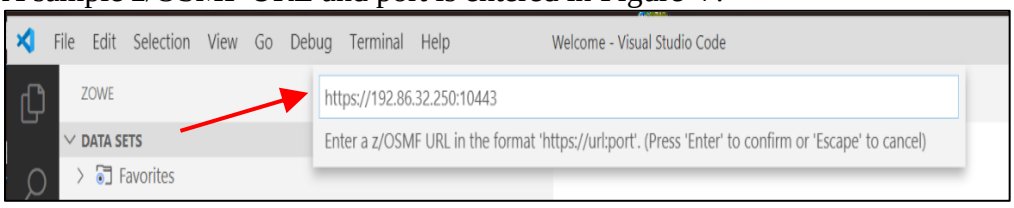

Figure 7. *Specified z/OSMF URL*

8. The connection prompts for Username as shown in Figure 8.

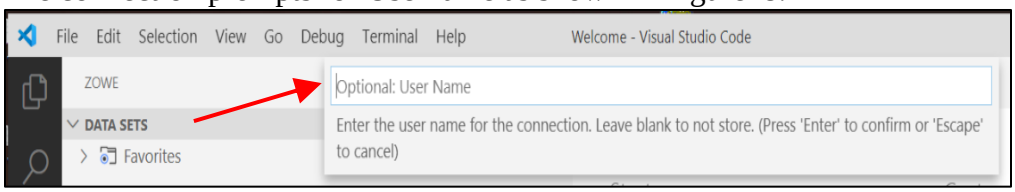

Figure 8. *User name prompt*

9. A sample username, Z99998, is entered as shown in Figure 9. You can find your ID in your account registration confirmation email.

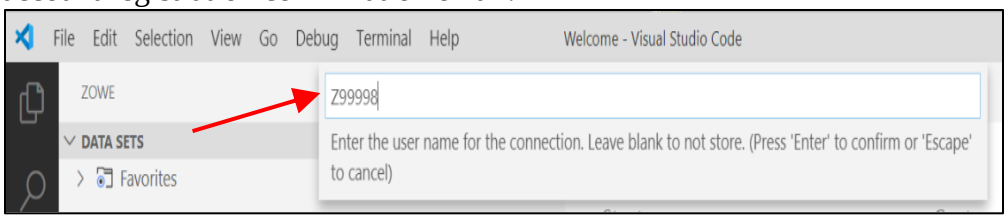

Figure 9. *Specified username*

10. The connection prompts for the Username Password as shown in Figure 10. You can find your username password in your account registration confirmation email.

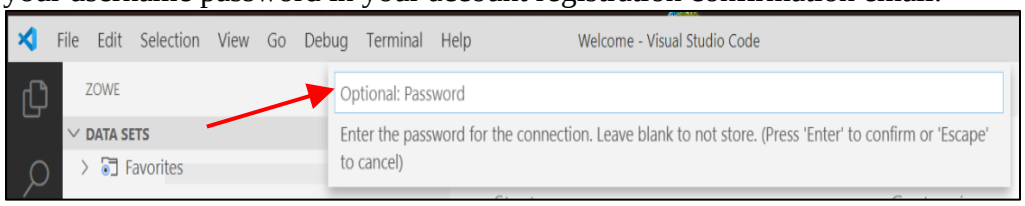

Figure 10. *Password prompt*

11. Enter the Username Password as shown in Figure 11.

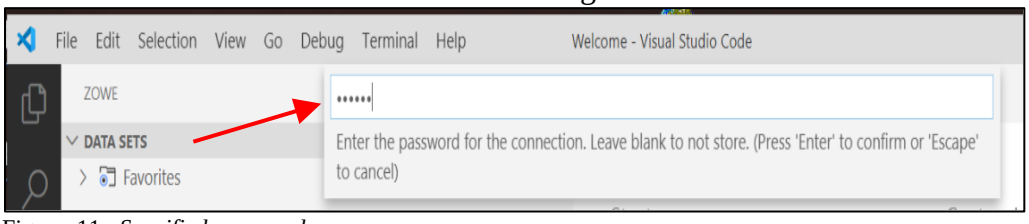

Figure 11. *Specified password*

12. Select **False – Accept connections with self-signed certificates** to authorize workstation connection as shown in Figure 12.

Figure 12. *Accept connections with self-signed certifications*

13. Result is Favorites in the Data Sets, Unix System Services, and Jobs sections as shown in Figure 13.

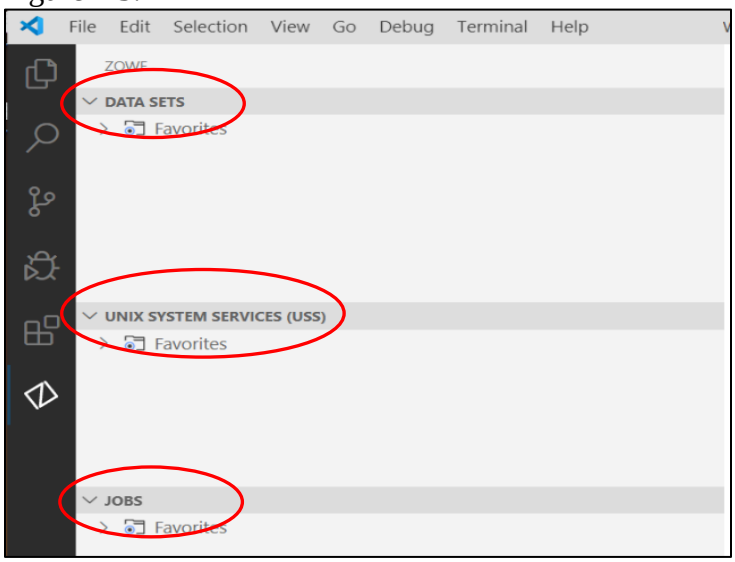

Figure 13. *Favorites*

14. Again, click on the + to the far right on the Data Sets selection. Result is another prompt to Create a New Connection to z/OS, the connection you created in step 5 is in the connection list. Select your connection for the Data Sets available to the previously defined , in our case LearnCOBOL, connection to z/OS as shown in Figure 14.

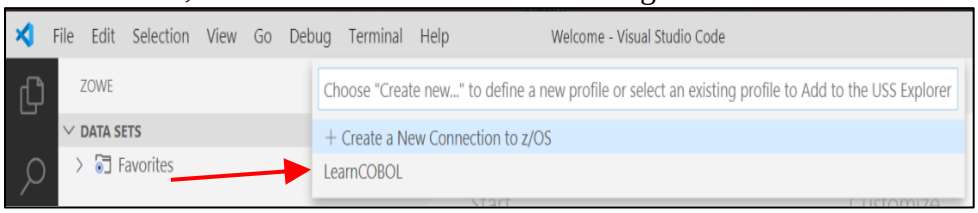

Figure 14. *LearnCOBOL connection*

15. Expansion of the connection (LearnCOBOL) reads “Use the search button to display datasets”. Click the search button as shown in Figure 15.

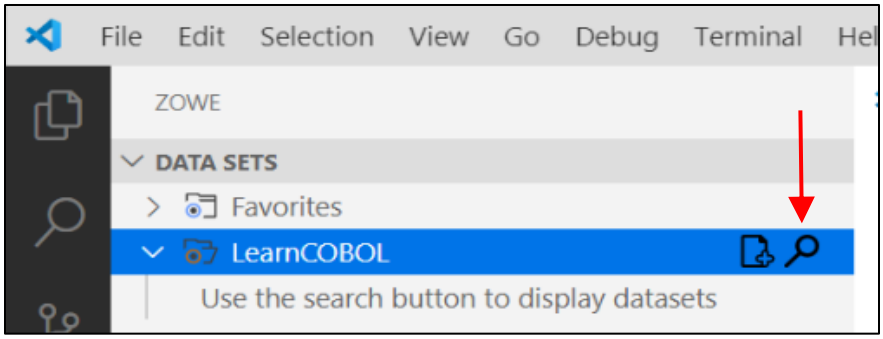

Figure 15. *Search button*

16. A prompt to "Select a filter" appears for ID Z99998. Select the + to 'Create a new filter" as shown in Figure 16.

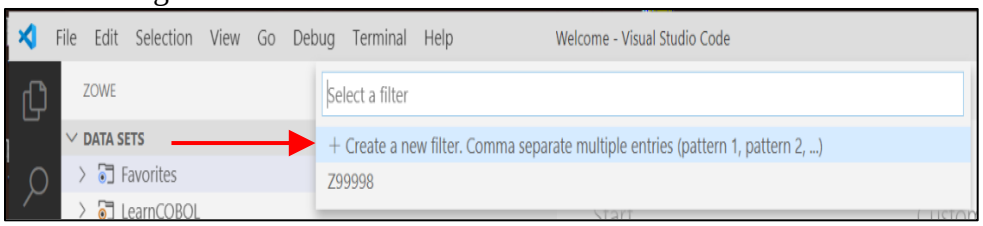

Figure 16. *Select a filter*

17. A prompt appears to enter the filter name to be searched as shown in Figure 17.

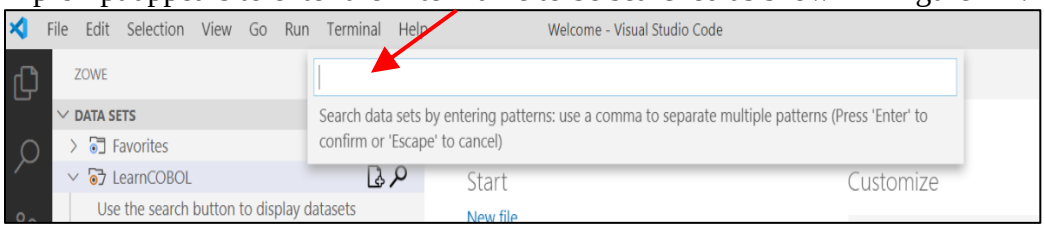

Figure 17. *Filter name to be searched*

18. ID Z99998 has lab data set names that begin the Z99998. Therefore, Z99998 is entered as the filter to searched for ID Z99998 as shown in Figure 18.

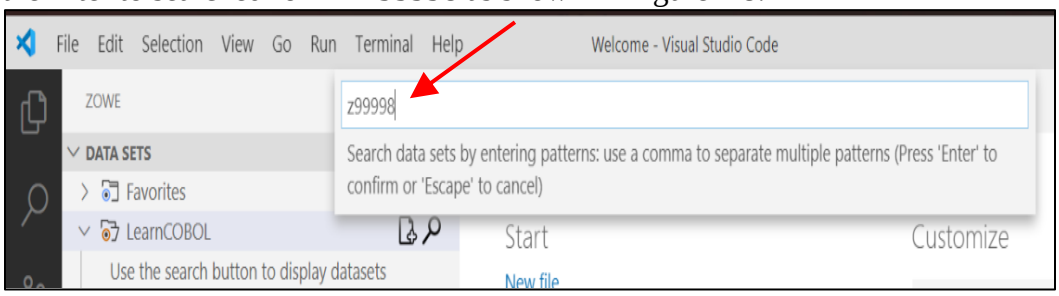

Figure 18. *Entered filter name*

19. A list of data set names beginning with Z99998 for ID Z99998 from z/OS Connection LearnCOBOL appears as shown in Figure 19.

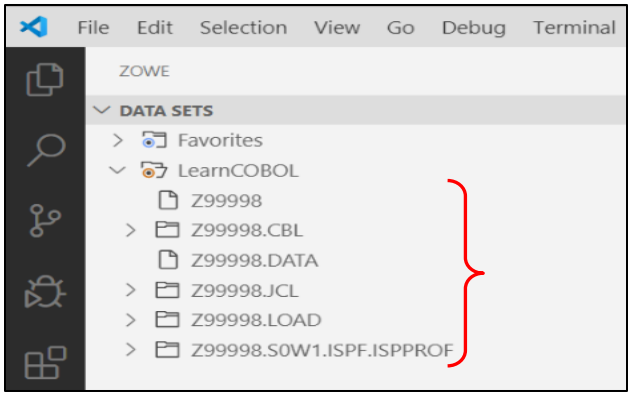

Figure 19. *Filtered data set names*

20. Expand **Z99998.CBL** to view COBOL source members, then select member **HELLO** to see a simple COBOL 'Hello World!' program as shown in Figure 20. 0.

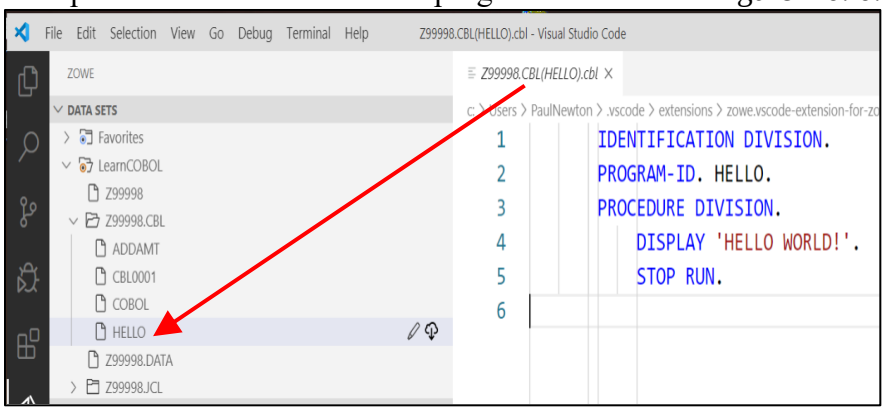

Figure 20. *Z9998.CBL*

21. Expand **Z99998.JCL** to view JCL and select member HELLO which is the JCL to compile and execute simple 'Hello World!' COBOL source code as shown in Figure 21.

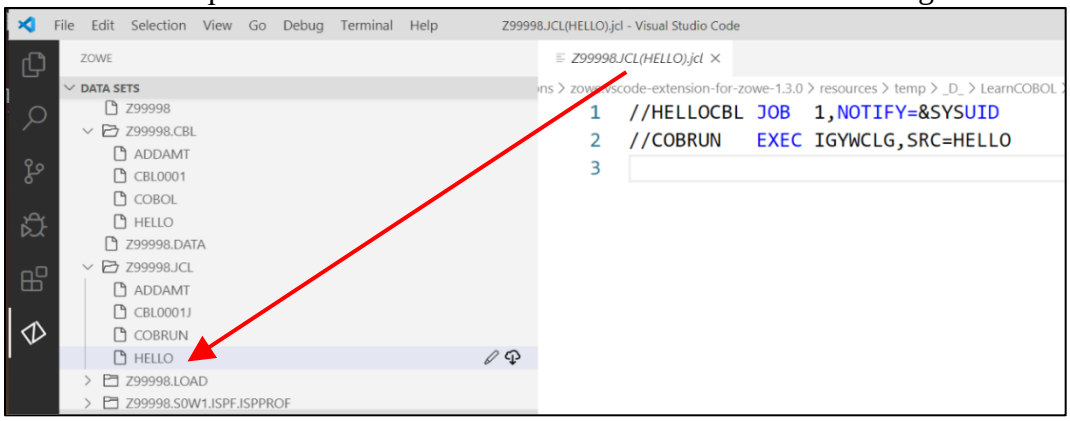

Figure 21. *Z99998.JCL*

22. Right click on JCL member **HELLO**, a section box appears. Select **Submit Job** for system to process HELLO JCL as shown in Figure 22. The submitted JCL job compiles the COBOL HELLO source code, then executes the COBOL HELLO program.

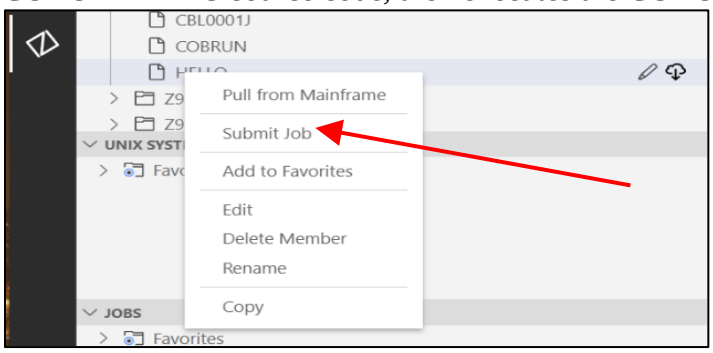

Figure 22. *Submit Job*

23. Observe the 'Jobs' section in Zowe Explorer as shown in Figure 23.

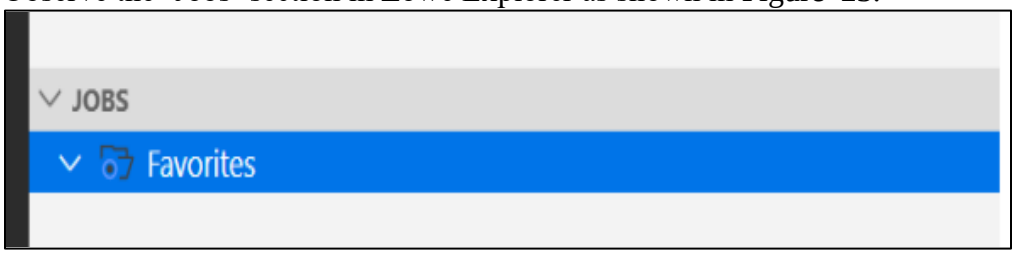

Figure 23. *JOBS section*

24. Again, click on the + to the far right on the Jobs selection. Result is another prompt to 'Create new'. Select your connection (LearnCOBOL) from the list as shown in Figure 24.

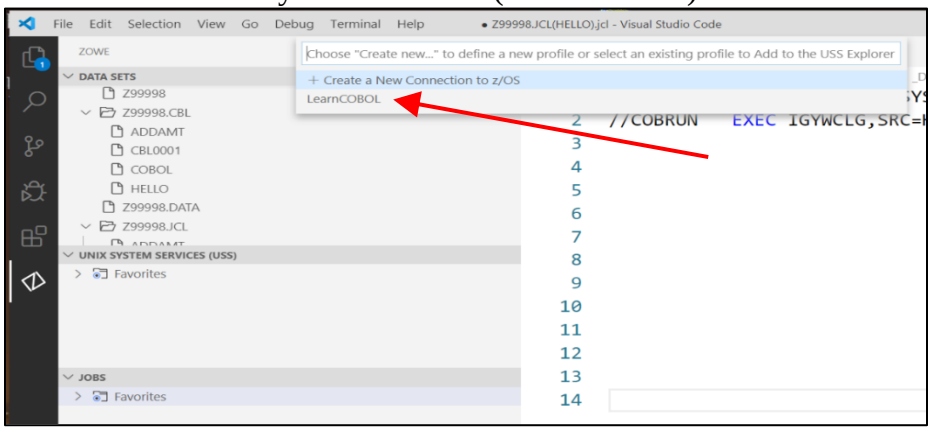

Figure 24. *Select LearnCOBOL connection*

25. As a result, the JCL jobs owned by ID Z99998 appear. HELLOCBL is the JCL job name previously submitted. Expand the **HELLOCBL** output to view sections of the output as shown in Figure 25.

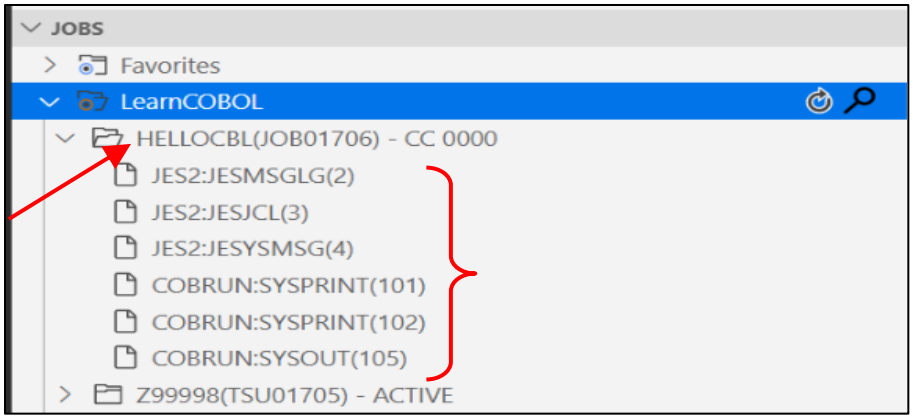

Figure 25. *HELLOCBL output*

26. Select **COBRUN:SYSPRINT(101)** to view the COBOL compiler output. Scroll forward in the COBOL compile to locate the COBOL source code compiled into an executable module as shown in Figure 26. Observe the Indicator Area in column 7, A Area beginning in column 8, and B Area beginning in column 12. Also, observe the period (.) scope terminators in the COBOL source.

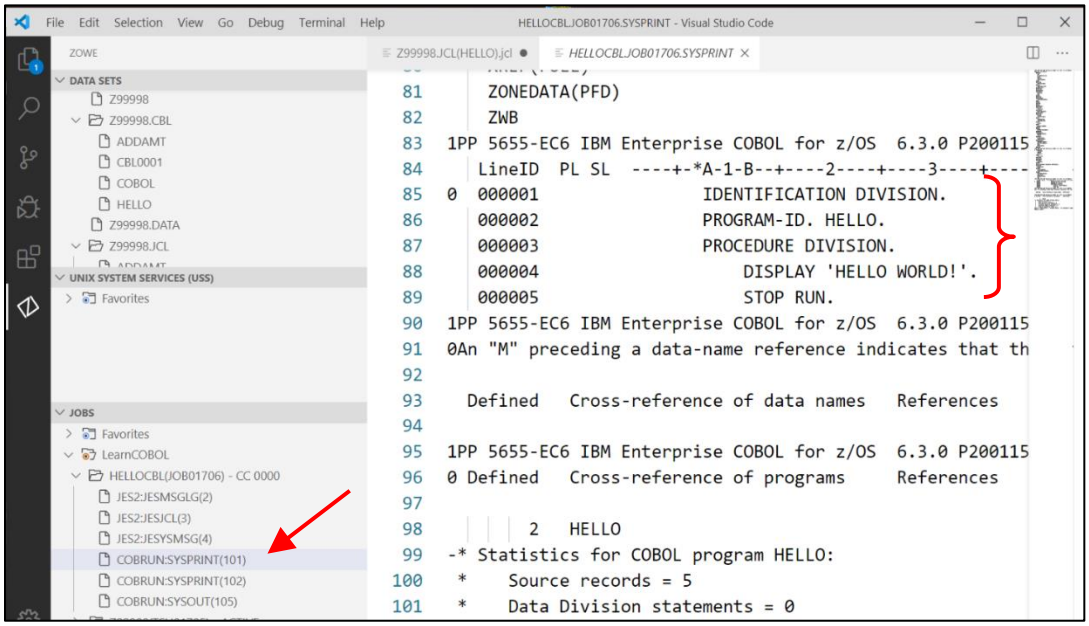

Figure 26. *COBOL compiler output*

27. View the COBOL program execution by selecting **COBRUN:SYSOUT(105)** from the LearnCOBOL in the Jobs section of Zowe Explorer as shown in Figure 27.

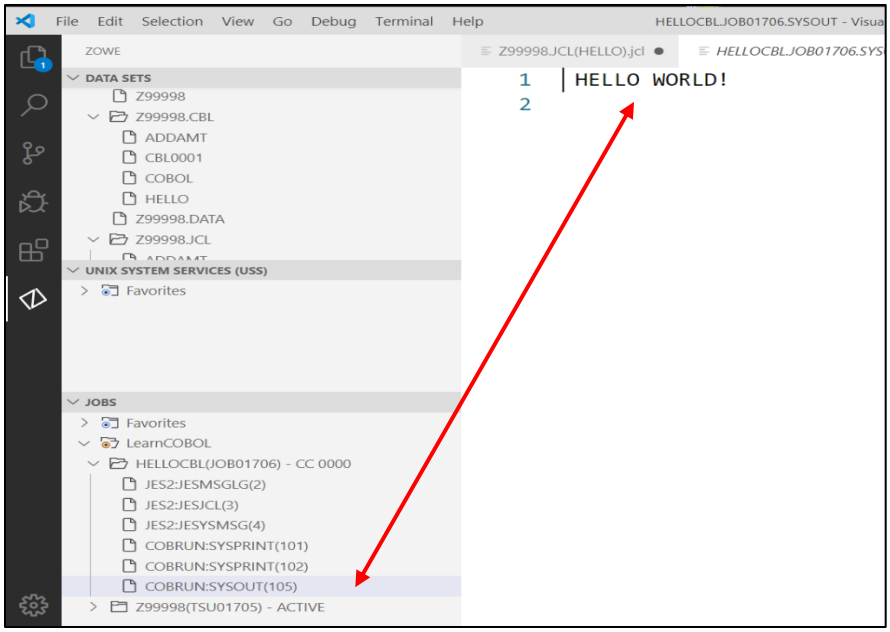

Figure 27. *COBOL program execution*

28. The following URL is another excellent document describing the above VSCode and Zowe Explorer details with examples:

    https://marketplace.visualstudio.com/items?itemName=Zowe.vscode-extension-for-zowe