
Lab 3. Successful vs. Unsuccessful Job Execution

Note: It may take a few seconds to load in all segments of this lab. If files are not loading, hit the refresh button on the list that appears when hovering over the section bar.

Overview

In this lab exercise you will run a simple payroll application that will result in a successful compile and execution. You will then run a second payroll application that will produce an error. You are tasked with identifying the issue, fixing it, and resubmitting the program for a successful compile and execution.

Objectives

- Submit and view output from successful run of PAYROL00 program
- Submit and view output from unsuccessful run of PAYROL0X program
- Identify the root of the error
- Fix the error
- Resubmit and view output from PAYROL0X job for a successful run

Lab instructions

1. View the PAYROL00 COBOL source code member in the 'id'.CBL data set.
2. Submit the JCL member, PAYROL00, from the id.JCL, where id is your provided username. This is where id.JCL(PAYROL00) compiles and successfully executes the PAYROL00 program.

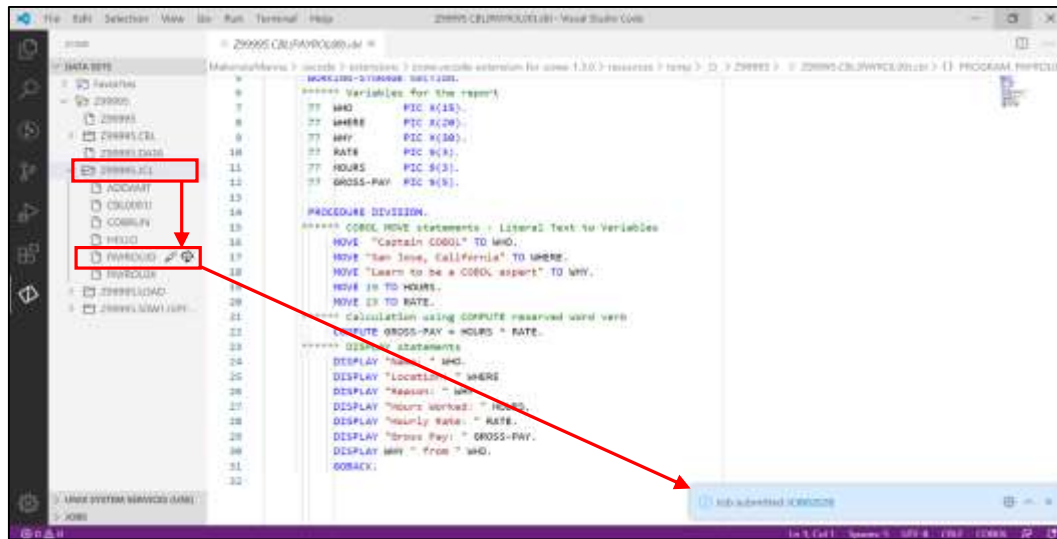


Figure 1. Submit PAYROL00 job

Note: If you receive this error message after submitting the job:



That is because you submitted the job from the .CBL data set and not the .JCL data set.

3. View both compile and execution of PAYROL00 job output, referenced in Figure 2.

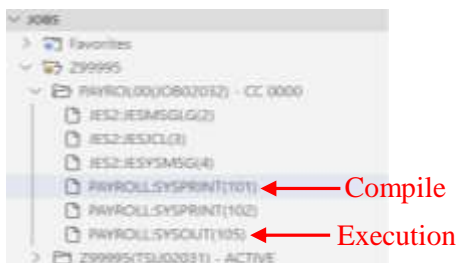


Figure 2. PAYROL00 output

4. Next, view PAYROL0X COBOL source code member in id.CBL data set.

5. View and submit the JCL member, PAYROL0X, from the id.JCL dropdown. This is where id.JCL(PAYROL0X) compiles and executes the PAYROL0X program.
6. View the compile of PAYROLL0X job output, notice there is no execution output.

Do you notice a difference between this compile and the previous job compile shown in Figure 3. 3?

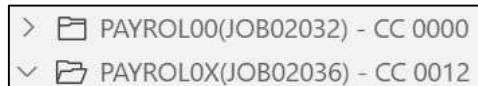


Figure 3. Compare job compiles

The difference is the return/condition code associated with each job output, located both next to the job output name within the JOBS section as shown above, or at the end of the compile output as, 0Return code ##. A return code of 12 means there was an error, but how do we know what that error was? Continue to find out!

7. Observe the text associated with IGYPA3146-S on line 137 within the job output (compile), illustrated in Figure 4.

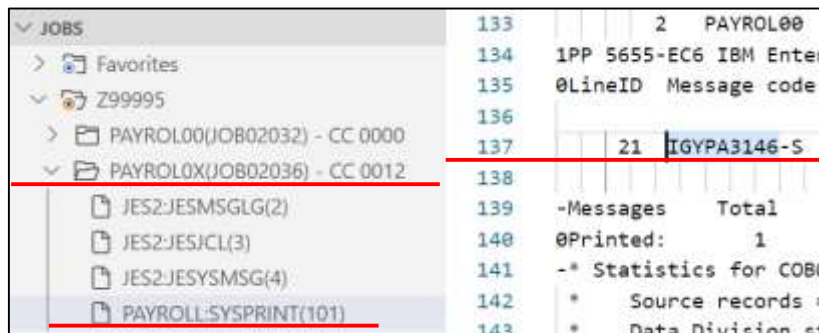


Figure 4. IGYPA3146-S message

Notice that this line tells you to focus on the GROSS-PAY picture clause in order to identify the problem. Use this information, modify the PAYROL0X COBOL source code to fix the error. Be sure you are editing the correct code.

8. After modifying, re-submit the PAYROL0X JCL to verify the problem has been identified and corrected, resulting in a successful compile and execution with a return code of zero, shown in Figure 5.

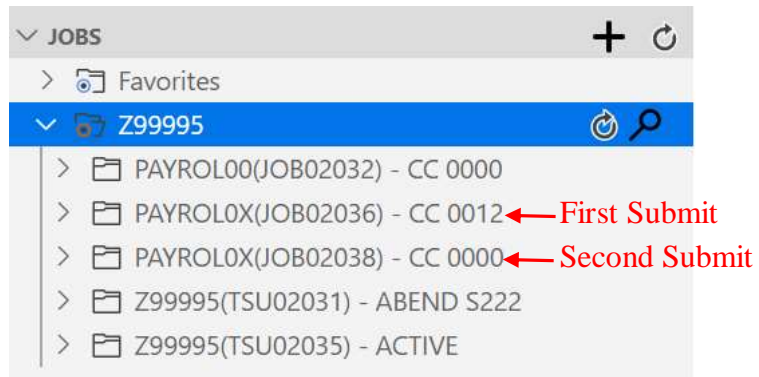


Figure 5. Compare return codes

Lab 4. Reading Records

Note: All action items followed by a * sign, include exercise hints, which are located at the end of this exercise.

Overview

This lab demonstrates the 'end-of-file' COBOL coding technique for reading all data records from a sequential file.

Objectives

- Familiarize with COBOL source code vs. JCL
- Identify error in program using the IGY message
- Correct the source code
- Analyze ABENDU4038 output
- Correct the JCL

Lab instructions

1. If not already, open VSCode and select Zowe Explorer from the left sidebar.

Note: If you are opening a new instance of VSCode (i.e. you closed out of it after the previous usage), you may need to 'Select a filter' again. You can do so by selecting the search icon (🔍) next to your named connection in the DATA SETS section and then reselecting the filter previously used. It should be in the listed filters after you have selected the search symbol.

2. View these COBOL source code members listed in the id.CBL data set:
 - CBL0001
 - CBL0002

3. View these three JCL members in the id.JCL data set:
 - CBL0001J
 - CBL0002J
 - CBL0003J

Figure 1. *Id.JCL(CBL0001J).jcl*

4. Submit job, JCL(CBL0001J), within the DATA SET section.
5. View that job output using the JOBS section.
 - COBRUN:SYSRINT(101) - COBOL program compiler output
 - RUN:PRTLINE(103) - COBOL program execution output, shown in Figure 2.

Line	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	17891797	\$10,000.00	\$188.74	WASHINGTON	George	longed to retire to his fields at Mount											
	17971801	\$10,000.00	\$3,188.33	ADAMS	John	retired to his farm in Quincy											
	18011809	\$10,000.00	\$7,008.13	JEFFERSON	Thomas	retired to Monticello											
	18091817	\$10,000.00	\$503.13	MADISON	James	retirement at Montpelier											
	18171825	\$10,000.00	\$31,313.13	MONROE	James	Russia must not encroach southward											
	18251829	\$10,000.00	\$31,250.33	ADAMS II	John Quincy	collapsed on the floor of the House fro											
	18291837	\$10,000.00	\$3,318.30	JACKSON	Andrew	that to the victors belong the spoils											
	18371841	\$10,000.00	\$325.00	VAN BUREN	Martin	independent treasury system											
	18411841	\$10,000.00	\$313.50	HARRISON	William Henry	ornate with classical allusions											
	18411845	\$10,000.00	\$121.65	TYLER	John	opposed the Missouri Compromise											
	18451849	\$10,000.00	\$314.05	POLK	James	health undermined from hard work											
	18491850	\$10,000.00	\$828.20	TAYLOR	Zachary	acted at times as though he were above											
	18501853	\$10,000.00	\$373.10	FILLMORE	Millard	signed the Fugitive Slave Act											
	18531857	\$10,000.00	\$315.07	PIERCE	Franklin	he tried to persuade Spain to sell Cuba											
	18571861	\$10,000.00	\$7.90	BUCHANAN	James	widening rift over slavery											
	18611865	\$100,000.00	\$313.13	LINCOLN	Abraham	new birth of freedom											
	18651869	\$100,000.00	\$603.14	JOHNSON	Andrew	freedmen were beginning to appear											

Figure 2. RUN:PRTLINE(103) for JCL(CBL0001J)

6. Submit job, JCL(CBL0002J), within the DATA SET section.
7. View that job output using the JOBS section.
 - COBRUN:SYSPRINT(101) - COBOL program compiler output

Locate COBOL compiler severe message IGYPS2121-S within the output file referred to in step 7, shown in Figure 3.

```
==000074== IGYPS2121-S "PRINT-REX" was not defined as a data-name. The statement was discarded.
```

Figure 3. IGYPS2121-S message

8. Edit CBL(CBL0002):
 - Determine appropriate spelling of PRINT-REX, correct it within the source code and save the updated source code.
9. Re-submit job, JCL(CBL0002J), using the DATA SET section and view the output in the JOBS section.
 - COBRUN:SYSPRINT(101) COBOL program compiler output
 - RUN:PRTLINE(103) is the COBOL program execution output (if correction is successful)
10. Submit job, JCL(CBL0003J), using the DATA SET section.
11. View CBL0003J ABENDU4038 output, using the JOBS section:

- View the IGVZ00355 abend message in RUN:SYSOUT(104) from the COBOL program execution output.
- IGVZ00355 reads, program is unable to open or close ACCTREC file name, shown in Figure 4. guiding you to the root of the error.



Figure 4. RUN:SYSOUT(104) message

12. Fix this error by editing JCL(CBL0003J):

- Determine the DDNAME needed, but missing or misspelled. *
- Correct it within the code and save

13. Re-submit job, JCL(CBL0003J), using the DATA SET section.

14. View CBL0003J output using the JOBS section, your output should look like Figure 5.

- RUN:PRTLINE - COBOL program execution output (if correction is successful)

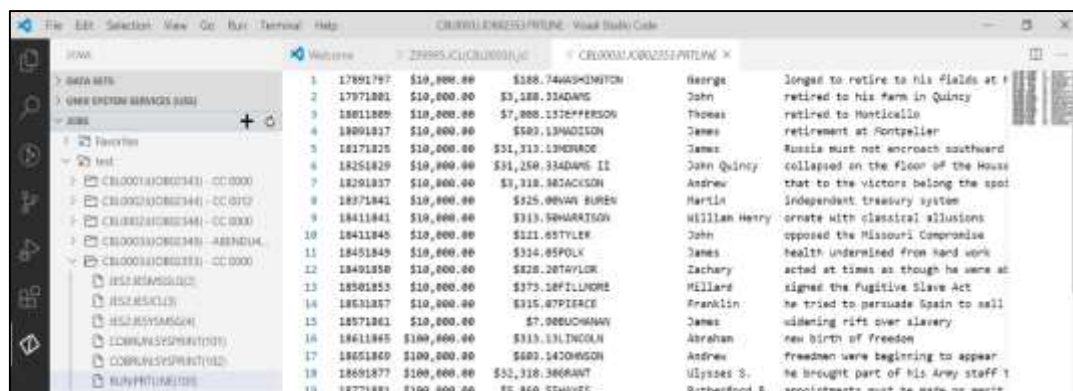


Figure 5. RUN:PRTLINE(103) for JCL(CBL0003J)

Lab hints

13. The error is located on line 11, adjust 'ACCTREX' accordingly.

10	//STEPLIB	DD DSN=&SYSUID..LOAD,DISP=SHR
11	//ACCTREX	DD DSN=&SYSUID..DATA,DISP=SHR
12	//PRTLINE	DD SYSOUT=*,OUTLIM=15000

Figure 6. Error in *id.JCL(CBL0003J).jcl*

Lab 5. Generating Reports

Overview

This lab utilizes two COBOL programs, CBL0004 and CBL0005, located within your id.CBL data set, as well as two JCL jobs, CBL0004J and CBL0005J, located within your id.JCL data set. The JCL jobs are used to compile and execute the COBOL programs, as discussed in previous labs. You will practice generating reports with COBOL, in addition to, reformatting displayed output for a report.

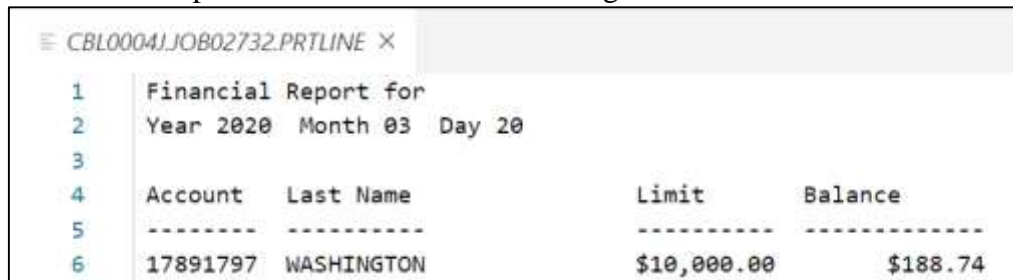
Objectives

- Submit job and view successfully generated report
- Compare the differences between two reports
- Edit the source code for the second report to reflect the proper format for currency
- Successfully resubmit the job with the changes reflected in the output

Lab instructions

Using VSCode and Zowe Explorer:

1. Submit job: CBL0004J
2. Observe the report written with headers like Figure 1. below.



The screenshot shows a terminal window with the title 'CBL0004J.JOB02732.PRTLINE X'. The output is a financial report with the following content:

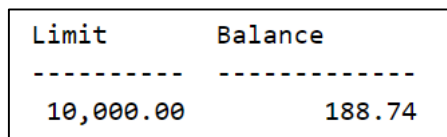
```

1 Financial Report for
2 Year 2020 Month 03 Day 20
3
4 Account      Last Name      Limit      Balance
5 -----
6 17891797    WASHINGTON    $10,000.00    $188.74

```

Figure 1. Report with headers

3. Submit job: CBL0005J
4. Observe the report data lines are written without dollar currency symbol, illustrated in Figure 2.



The screenshot shows a terminal window with the title 'CBL0005J.JOB02734.PRTLINE X'. The output is a financial report with the following content:

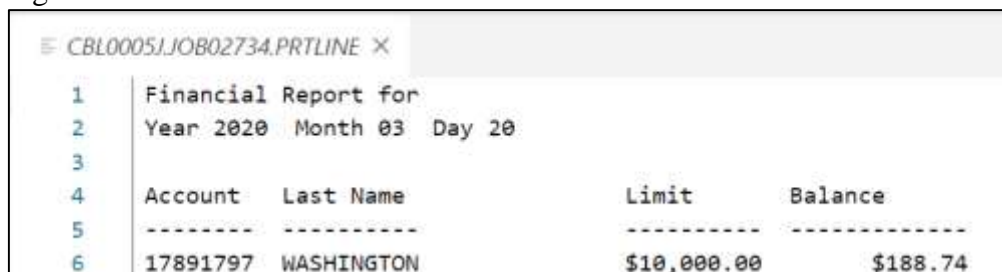
```

Limit      Balance
-----
10,000.00    188.74

```

Figure 2. No currency symbol in output

5. Modify id.CBL(CBL0005) to include the dollar currency symbol in the report.
**Hint: Compare with CBL0004 line 25*
6. Re-submit job: CBL0005J
7. Observe the report data lines should now include the dollar currency symbol, shown in Figure 3.



The screenshot shows a terminal window with the title 'CBL0005J.JOB02734.PRTLINE X'. The output is a financial report with the following content:

```

1 Financial Report for
2 Year 2020 Month 03 Day 20
3
4 Account      Last Name      Limit      Balance
5 -----
6 17891797    WASHINGTON    $10,000.00    $188.74

```

Figure 3. Currency symbol added to output