
Lab 6. Editing Reports

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

Overview

This lab requires two COBOL programs, CBL0006 and CBL0007 and two respective JCL Jobs, CBL0006J and CBL0007J, to compile and execute the COBOL programs. All of which are provided to you in your VSCode - Zowe Explorer.

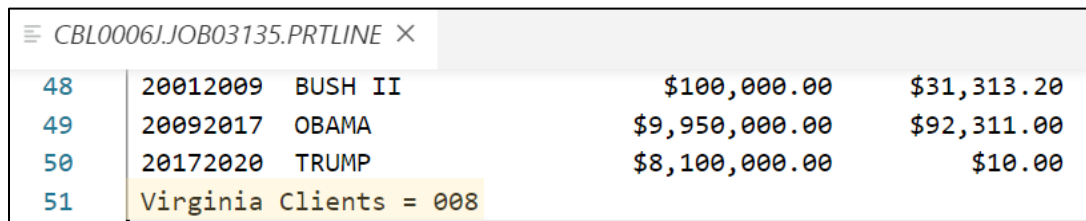
Objectives

- Observe new additions to the report
- Submit job and verify the additions are present in the generated report
- Identify and correct syntax error
- Verify the correction is successful

Lab instructions

Using VSCode and Zowe Explorer:

1. Take a moment and look over the source code of the two COBOL programs provided: CBL0006 and CBL0007.
2. Compare CBL0006 with CBL0005 from the previous lab. Do you notice the differences?
 - a. Observe the new CLIENTS-PER-STATE line within the WORKING-STORAGE > PROCEDURE DIVISION.
 - b. Observe the new paragraph IS-STATE-VIRGINIA within that same division.
 - c. This paragraph checks whether the client is from Virginia. If that condition is met (true) then the program should add 1 to the clients from Virginia total.
 - d. Program writes “Virginia Clients = “, in last line of report.
3. Submit CBL0006J
4. View the job output from the JOBS section and verify the steps mentioned above were executed.



Line	Date	Name	Amount	Total
48	20012009	BUSH II	\$100,000.00	\$31,313.20
49	20092017	OBAMA	\$9,950,000.00	\$92,311.00
50	20172020	TRUMP	\$8,100,000.00	\$10.00
51	Virginia Clients = 008			

Figure 1. *Id.JCL(CBL0006J) output*

5. Submit CBL0007J
6. Find the compile error, IGYPS2113-E, in the job output.
7. Go ahead and modify id.CBL(CBL0007) to correct the syntax error outlined by the IGYPS2113-E message.*
8. Re-submit CBL0007J
9. Validate that the syntax error was corrected by getting an error free output file.

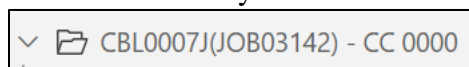


Figure 2. *Successful compile*

Lab Hints

7.

```
IS-STATE-VIRGINIA.  
    IF USA-STATE = 'Virginia' THEN  
        ADD 1 TO VIRGINIA-CLIENTS  
  
    END-IF.
```

Lab 7. Implementing Trailers

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

Overview

This lab requires two COBOL programs, CBL0008 and CBL0009 and two respective JCL Jobs, CBL0008J and CBL0009J, to compile and execute the COBOL programs. All of which are provided to you in your VSCode - Zowe Explorer.

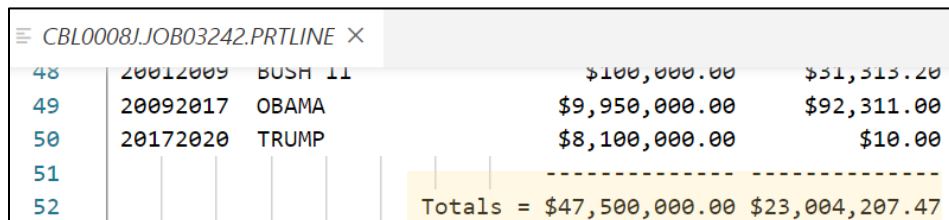
Objectives

- Practice the use of trailers
- Identify data name definition error using the IGY message
- Correct the source code
- Verify the correction is successful

Lab instructions

Using VSCode and Zowe Explorer:

1. Take a moment and look over the source code of the two COBOL programs provided: CBL0008 and CBL0009.
2. Submit CBL0008J
3. Observe report written with trailers consisting of limit and balance totals at the bottom of the output.



48	20012009	BUSH 11	\$100,000.00	\$31,313.20
49	20092017	OBAMA	\$9,950,000.00	\$92,311.00
50	20172020	TRUMP	\$8,100,000.00	\$10.00
51				
52			Totals = \$47,500,000.00	\$23,004,207.47

Figure 1. Limit and balance totals

4. Submit CBL0009J
5. Was the job successful? If not, find the compile error message to understand why.
6. Modify id.CBL(CBL0009), correcting the compile error.*

```
==000141==> IGYPS2121-S "TLIMIT" was not defined as a data-name. The statement was discarded.
```

Figure 2. IGYPS2121-S error message

7. Re-submit CBL0009J
8. Validate that the syntax error was corrected by getting an error free output file like in Figure 3. The correction should report written with trailers consisting of limit and balance totals, like Figure 1.

```
> CBL0009J(JOB03243) - CC 0012
> CBL0009J(JOB03245) - CC 0000
```

Figure 3. Successful compile

Lab Hints

6.

```
01 TLIMIT-TBALANCE.  
05 TLIMIT          PIC S9(9)V99 COMP-3 VALUE ZERO.  
05 TBALANCE        PIC S9(9)V99 COMP-3 VALUE ZERO.
```

Lab 8. Working with COMP-3

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

Overview

Many of the previous COBOL lab programs you have worked with thus far are reading records containing two packed decimal fields, the client account limit and the client account balance. In this lab, the total of all client account limits and balances used a COMPUTE statement, where the COMP-3 fields contained the packed decimal internal data.

Objectives

- Practice the use of COMPUTE
- Understand the need for COMP-3 fields
- Identify the compile error and understand the underlying cause
- Use that information to correct and verify the program runs successfully

Lab instructions

Preface

- **What happens when an internal packed decimal field is not described using COMP-3?**

Without using COMP-3 to describe the field, the COBOL program treats the data as DISPLAY data (EBCDIC format). This lab demonstrates what happens during program execution without using COMP-3.

Using VSCode and Zowe Explorer:

1. Submit the job, id.JCL(CBL0010J)
2. Observe that the compile of the COBOL source was successful, however, also observe that the execution of the job failed. How can you tell?

There's no CC code next to CBL0010J(JOB#), instead there is an ABENDU4038 message. U4038 is a common user code error typically involving a mismatch between the external data and the COBOL representation of the data.

3. Read the execution SYSOUT message carefully. The SYSOUT message mistakenly believes the records are 174 characters in length while the program believes the records are 170 characters in length.

Packed decimal (COMP-3) expands into two numbers where only one number would typically exist. If the program reads a packed decimal field without describing the field as COMP-3, then program execution becomes confused about the size of the record because the PIC clause, S9(7)V99, is expecting to store seven numbers plus a sign digit when only three word positions are read. Therefore, execution reports a four-record length position discrepancy.

4. Edit id.CBL(CBL0010) to identify and correct the source code problem.*
5. Submit id.JCL(CBL0010J) and verify correction is successful with a CC 0000 code.

Lab Hints

4. The ACCT-LIMIT PIC clause in the ACCT-FIELDS paragraph should be the same as the PIC clause for ACCT-BALANCE.

Lab 9. Intrinsic Functions

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

Overview

This lab contains data that includes a last name, where last name is all upper-case. It demonstrates the use of intrinsic functions together with reference modification to lower-case the last name characters, except the first character of the last name. This lab requires two COBOL programs, CBL0011 and CBL0012 and two respective JCL Jobs, CBL0011J and CBL0012J, to compile and execute the COBOL programs. All of which are provided to you in your VSCode - Zowe Explorer.

Objectives

- Practice the manipulation of data expression in output
- View and understand the usage of intrinsic functions paired with reference modifications
- Correct the syntax for the date/time intrinsic function
- Verify the change is valid

Lab instructions

Using VSCode and Zowe Explorer:

1. Submit job, CBL0011J.
2. Observe the report output, last name, with first character upper-case and the remaining characters lower-case.

Figure 1, below, illustrates the difference in output from lab 8 compared to this lab. Notice that in the previous lab, the last names were listed in all capitalized characters, whereas, as previously stated, this lab output has only the first character of the last name capitalized.

Last Name	Last Name
-----	-----
Washington	WASHINGTON
Adams	ADAMS
Jefferson	JEFFERSON
Lab 9	Lab 8

Figure 1. Lab 9 vs. lab 8 output

3. Observe the PROCEDURE DIVISION intrinsic function, lower-case, within the WRITE-RECORD paragraph. This intrinsic function is paired with a reference modification resulting in output of last name with upper-case first character and the remainder in lower-case.
4. Submit CBL0012J
5. Observe the compile error.
Previous lab programs made use of a date/time intrinsic function. The date/time intrinsic function in this lab has a syntax error that needs to be identified and corrected.
6. Modify id.CBL(CBL0012) correcting compile error.*
7. Re-submit CBL0012J
8. Corrected CBL0012 source code should compile and execute the program successfully. A successful compile will result in the same output as CBL0011J.

Lab Hints

6. Refer to CBL0011 line 120 for the proper formatting of the function-name causing the compile error.

