Extended

230222-1424

PLI1

Introduction to building and running PL/I programs

Z Xplore

THE CHALLENGE

PL/1 (Programming Language One) is a procedural, imperative computer programming language developed and first published in 1964 by IBM. It is designed for scientific, engineering, business and system programming. It has been used by academic, commercial and industrial organizations since it was introduced, and is still one of the most used enterprise programming languages today.

During this challenge you will work with the basics of the PL/I programming language, and the compiling and running of the code using JCL.

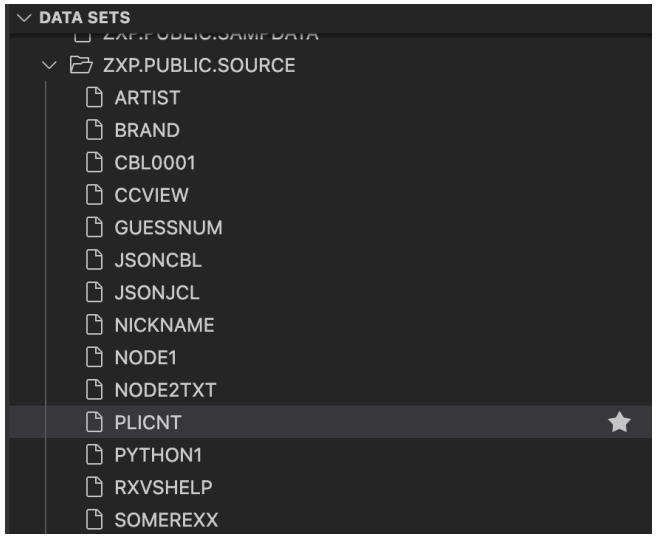
INVESTMENT

Steps	Duration
5	60 minutes

PLI1|230222-1424

1 UNDERSTANDING THE FORMAT

Use VSCode to logon to the IBM Z XPlore platform. Set a Data Sets filter using **ZXP.PUBLIC** , locate the dataset **'ZXP.PUBLIC.SOURCE'** , and open the member **PLICNT**.



PL/1 location

Take a look at the code and see how the program is structured.

```
1   PLICNT: PROC OPTIONS(MAIN);                                          00010001
2   DCL PRTLINE FILE RECORD OUTPUT ENV(FB,RECSIZE(7));                   00020001
3   DCL PRTDONE FILE RECORD OUTPUT ENV(FB,RECSIZE(80));                  00030000
4   DCL SYSPRINT FILE STREAM PRINT;                                      00040000
5   DCL PRTLINE_RCD PIC'9999999';                                        00050001
6   DCL 1 PRTDONE_RCD1,                                                  00051000
7         2 PRTDONE_COMMENT CHAR(80)                                     00052002
8           INIT('PL/1 program counted from 1 to 1,000,000');            00052102
9   DCL I     FIXED DEC(7) INIT(0);                                      00053002
10  DCL COUNT FIXED DEC(7) INIT(0);                                      00150001
11  OPEN FILE(PRTLINE);                                                  00160002
12  DO I=1 TO 1000000;                                                   00270002
13     COUNT = COUNT + 1;                                                00280000
14  END;                                                                 00310000
15  PRTLINE_RCD = COUNT;                                                 00320001
16  WRITE FILE(PRTLINE) FROM (PRTLINE_RCD);                              00330001
17  WRITE FILE(PRTDONE) FROM (PRTDONE_RCD1);                             00350000
18  CLOSE FILE(PRTDONE);                                                 00360000
19  CLOSE FILE(PRTLINE);                                                 00370000
20  END PLICNT;                                                          00380001
21
```

PL/1 source code

Line 1

This is the name of the program without external dependencies and therefore defined as **MAIN**.

Line 2-10

The DECLARE statement declares session variables. In this program you can see the declaration of working variables as well as external files (**PRTLINE**, **PRTDONE**, **SYSPRINT**). Please notice the dataset attributes requested in these declarations for **PRTDONE** and **PRTLINE**. You will be taking a closer look at this later.

Line 11-14

This is the actual computation that this program will perform. The program will simply count from 1 to 1000000.

Line 16-20

Here the result is processed. As you can see, the output is transferred to the file variables, **PRTDONE** and **PRTLINE**.

2 PROCESSING USING JCL

In order to compile and execute this program, you will use a JCL script.

There is an example job definition in **‘ZXP.PUBLIC.JCL(PLICNT)’**.

Locate this JCL and copy it to your own JCL dataset for editing.

Take a look at the JCL file:

```
1  //PLICNT   JOB 1,NOTIFY=&SYSUID
2  //PLI      EXEC PGM=IBMZPLI,
3  //    PARM='OBJECT,OPTIONS,SOURCE',
4  //    REGION=0M
5  //STEPLIB  DD  DSN=IEL530.SIBMZCMP,DISP=SHR
6  //         DD  DSN=CEE.SCEERUN,DISP=SHR
7  //SYSIN    DD  DSN=ZXP.PUBLIC.SOURCE(PLICNT),DISP=SHR
8  //SYSPRINT DD  SYSOUT=*
9  //SYSOUT   DD  SYSOUT=*
10 //SYSLIN   DD  DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSALLDA,
11 //             SPACE=(CYL,(1,1)),DCB=(LRECL=80,BLKSIZE=3200)
12 //SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSALLDA,
13 //             SPACE=(1024,(200,50),,CONTIG,ROUND),DCB=BLKSIZE=1024
14 //*****************************************************/
15 // IF RC < 8 THEN
16 //*****************************************************/
17 //BIND     EXEC PGM=IEWBLINK,
18 //              PARM='XREF,COMPAT=PM3',REGION=2048K
19 //SYSLIB   DD  DSN=CEE.SCEELKED,DISP=SHR
20 //SYSPRINT DD  SYSOUT=*
21 //SYSLIN   DD  DSN=*.PLI.SYSLIN,DISP=(OLD,DELETE)
22 //SYSLMOD  DD  DSN=&SYSUID..LOAD(PLICNT),DISP=SHR
23 //SYSDEFSD DD  DUMMY
24 //SYSIN    DD  DUMMY
25 // ELSE
26 // ENDIF
27 //*****************************************************/
28 // IF RC < 5 THEN
29 //*****************************************************/
30 //EXECUTE  EXEC PGM=PLICNT,REGION=32M
31 //STEPLIB   DD DSN=&SYSUID..LOAD,DISP=SHR
32 //          DD DSN=CEE.SCEERUN,DISP=SHR
33 //SYSPRINT  DD SYSOUT=*
34 //CEEDUMP   DD DUMMY
35 //SYSUDUMP  DD DUMMY
36 //SYSOUT    DD SYSOUT=*,OUTLIM=10
37 //PRTDONE   DD SYSOUT=*,OUTLIM=10
38 //PRTLINE   DD DSN=&SYSUID..PLICNT.PRTLINE,DISP=OLD
39 //*****************************************************/
40 // ELSE
41 // ENDIF
```

PL/I build JCL

Line 1-13

This part of the JCL is where the compiler libraries are defined as well as other declarations needed to run the JCL.

The PL/I compiler uses quite a lot of memory - to support this, the requested memory allocation is defined in Line 4 - 0M means “as much as is needed”.

PLI1|230222-1424

Line 5 (**STEPLIB**) defines the location of the compiler library; line 7 (SYSIN) is where the source program is located.

Line 10 (**SYSLIN**) declares that the compiled output object will be available in a temporary dataset (deleted after the job finishes).

Line 14-26

The next step is executed after the compilation step (**PLI**) has succeeded. Line 15 checks for a return code from the PLI step of less than 8 - most traditional mainframe programs use a convention of setting the return code:

0	Success
4	Warnings, generally OK
8	Errors, output not usable
12	Problems with inputs; output not usable
16	Not possible to process

You can see that it should run the program **IEWBLINK** on Line 17. (This is the same program used in the jobs to compile and run COBOL progams)

Line 16-27

Here the program **IEWBLINK** is being called which is being used to “bind” the program. The Program Management Binder (also known as the “Linker”) takes the object from the SYSLIN dataset from the PLI complie step, binds it to the current system libraries, and places resulting the program module into the dataset defined by the **SYSLMOD** DD statement with the name ‘Zxxxxx.LOAD(PLICNT)’

Line 27-42

Here is where your freshly compiled program will being executed.

You can see that the output file PRTDONE is written into a joblog dataset.

On Line 39 you can see that the output of PRTLINE is being written in a sequential dataset called **‘Zxxxxx.PLICNT.PRTLINE’**.

Remember this, as it will help you later.

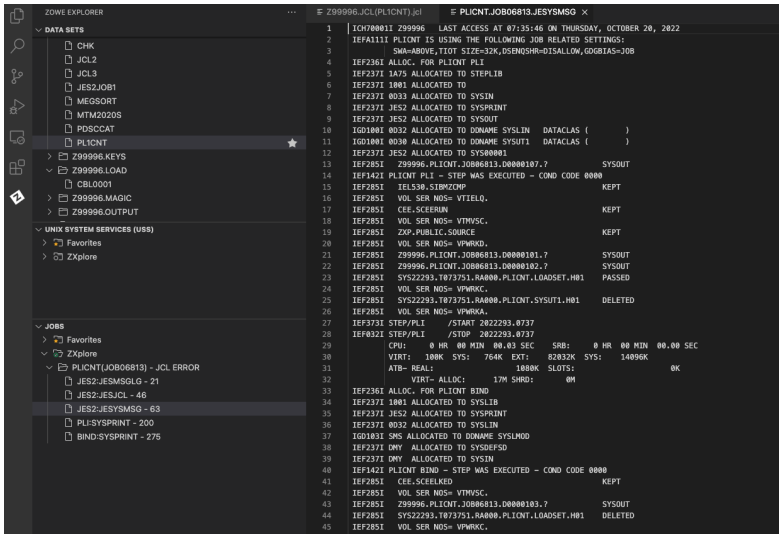
3 EXECUTE AND … FAIL …

Before making any changes, try to submit the **PLICNT** JCL and see what happens.

(Spoiler alert: it will fail!)

Use the joblog files to understand the error and try to fix it.

Tip: The error can be found in the **JESYSMSG** file.



PL/I build joblog

4 FIX AND TRY AGAIN

You have probably found the error which seems to be easy to solve.

But there is still one thing to take in consideration.

When you look at the PL/I code, you will see some details is the declaration of the variables.

Take a close look and see what attributes are expected in order to match the program file, the JCL DD, and the output dataset.

This will bring back memories of the Fundamentals JCL1 challenge, and will test your skills in allocating datasets with a particular “shape”.

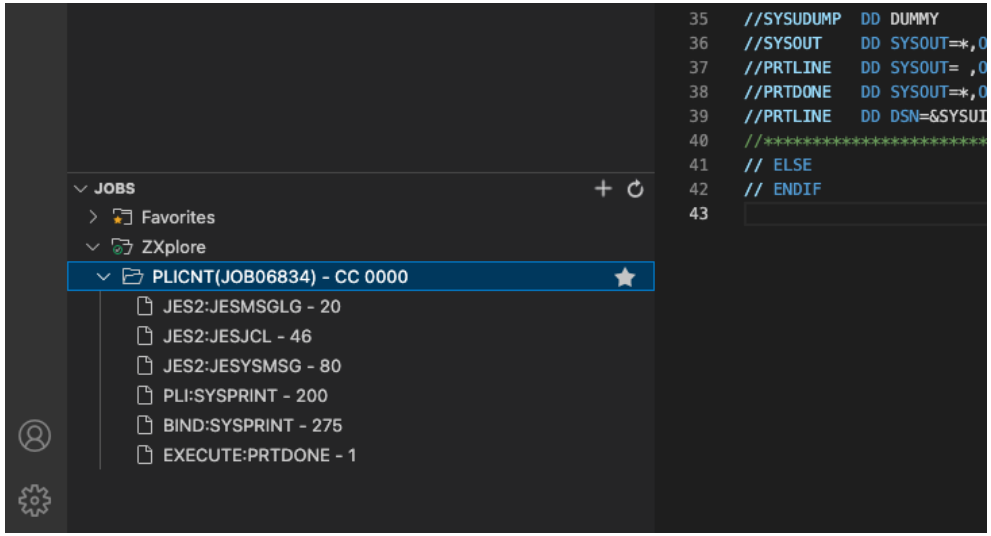
Here is a small hint if you really cannot find it:

https://ibm.biz/zxplore_extended_pl1

Allocate any required datasets and submit your PLICNT JCL again.

If all goes well, you see that your job did run succesfully and the job completion code is 0000.

Now you know where to look to find the output of your first PL/I program as well, Congratulations!



PL/I joblog

PLI1|230222-1424

5 VALIDATION

In order to complete this challenge and receive your credits, please submit **CHKPLI1** JCL which you can find in the **'ZXP.PUBLIC.JCL'** dataset.

The validation will check for the correct output (assuming no changes to the PLI progam source) in the correct dataset location.

Make sure that the dataset attributes have been configured to match the PLI program requirements.