

Tutorial 2

Erstellen, Kompilieren und Ausführen eines PLI-Programms

Copyright © Institut für Informatik, Universität Leipzig
ph v/2010/03

Ziele dieses Tutorials sind, ein PLI-Programm zu schreiben, ein passendes JCL-Script zu entwerfen, dass dieses PLI-Programm compilieren kann sowie für beides zuvor eine passende Entwicklungsumgebung zu schaffen.

Dieses Tutorial behandelt eine ähnliche Problematik wie die anderen Versionen des Tutorials 2. Alle diese Versionen unterscheiden sich im Kern in der Programmiersprache des zu erstellenden, compilierenden sowie auszuführenden Programms. Hier ist diese Programmiersprache PLI - in den anderen Versionen sind diese C, Assembler sowie Cobol. Auf eine detaillierte Beschreibung, wie die Aufgaben dieses Tutorials zu lösen sind, wird hier verzichtet. Nutzen Sie bei Bedarf das Tutorial 2 in der C-Version, die anderen Tutorials sowie das Tutorial "Fehlersuche in z/OS und z/OS-Anwendungen".

Aufgabe: Beschäftigen Sie sich mit dem nachfolgenden Beispiel des Erstellens, Compilierens sowie Ausführens eines PLI-Programms. Vollziehen Sie alle Schritte nach.

1. Einrichten der Entwicklungsumgebung

Als erstes sind drei Datasets anzulegen (allocate):

- Ein Dataset "PRAKT20.TUT2PLI.PLI02", der verschiedene PLI - Programme aufnehmen kann.
- Ein Dataset "PRAKT20.TUT2PLI.SCRIPTE" der verschiedene JCL-Scripte aufnehmen kann.
- Ein Dataset "PRAKT20.TUT2PLI.LOAD", der verschiedene ausführbare Programm-Module aufnehmen kann.

Aufgabe: Legen Sie die drei Datasets an. Benutzen Sie dabei die in der Abbildung 1 dargestellten Parameter. Die Datasets PRAKT20.TUT2PLI.PLI02 und PRAKT20.TUT2PLI.SCRIPTE sollten im Record format "FB", PRAKT20.TUT2PLI.LOAD im Record format "U" angelegt werden.

```

Menu  RefList  Utilities  Help
-----
                          Allocate New Data Set
                          More:      +
Data Set Name . . . . : PRAKT20.TEST.DATASET

Management class . . . . . (Blank for default management class)
Storage class . . . . . (Blank for default storage class)
Volume serial . . . . . (Blank for system default volume) **
Device type . . . . . (Generic unit or device address) **
Data class . . . . . (Blank for default data class)
Space units . . . . . KILOBYTE (BLKS, TRKS, CYLS, KB, MB, BYTES
                               or RECORDS)
Average record unit . . . . . (M, K, or U)
Primary quantity . . . 16 (In above units)
Secondary quantity . . . 1 (In above units)
Directory blocks . . . 2 (Zero for sequential data set) *
Record format . . . . .
Record length . . . . . 80
Block size . . . . . 320
Data set name type : PDS (LIBRARY, HFS, PDS, or blank) *
                    (YY/MM/DD, YYYY/MM/DD)

Command ==>
F1=Help      F3=Exit      F10=Actions  F12=Cancel
    
```

Abbildung 1: Allocate-Parameter

2. Erstellen des PLI-Programms

Ein PLI-Programm kann unter Nutzung des ISPF-Editors erstellt werden. Die Abbildung 2 zeigt ein Beispielprogramm.

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          PRAKT20.TUT2PLI.PLI02(V1) - 01.05          Columns 00001 00072
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001  PLI02: proc reorder options(main);
000002      dcl printrec char(80);
000003      printrec = 'Hallo Welt, unser erstes TSO-Programm in PLI';
000004      put  file(sysprint) list (printrec);
000005  end PLI02;
***** ***** Bottom of Data *****

Command ==>
F1=Help      F3=Exit      F5=Rfind     F6=Rchange  F12=Cancel

Scroll ==> PAGE

```

Abbildung 2: Beispiel-Programm in PLI

3. Erstellen des JCL-Scriptes

Ein PLI-Programm kann mittels eines JCL-Scriptes compiliert werden. Solch ein JCL-Script zeigt Abbildung 3. Man startet dieses mit "SUB" (Submit). Eine erfolgreiche Abarbeitung wird mit einer Nachricht ähnlich

"... JOB18585 \$HASP165 PRAK100D ENDED AT N1 MAXCC=0 CN(INTERNAL)"

beendet. Im Dataset "PRAKT20.TUT2PLI.LOAD" ist jetzt ein Member "PLI02" zu finden; also ein Member, das den gleichen Namen trägt wie das PLI-Programm.

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          PRAKT20.TUT2PLI.SCRIPTE(V1) - 01.04          Columns 00001 00072
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000010 //PRAKT20D JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
000020 //          REGION=4M
000030 //STEP1 EXEC IEL1CL,LIBPRFX=CEE
000040 //PLI.SYSIN DD DSN=&SYSUID..TUT2PLI.PLI02(V1),DISP=SHR
000050 //LKED.SYSLMOD DD DSN=&SYSUID..TUT2PLI.LOAD,DISP=SHR
000060 //LKED.SYSIN DD *
000070 NAME PLI02(R)
000080 /*
***** ***** Bottom of Data *****

Command ==> SUB          Scroll ==> PAGE
F1=Help      F3=Exit      F5=Rfind      F6=Rchange  F12=Cancel

```

Abbildung 3: JCL-Script

4. Ausführen des PLI-Programms

Vom "ISPF Primary Option Menu"-Panel aus läßt sich nun das PLI-Programm durch Eingabe des folgenden Kommandos starten:

```
TSO CALL 'PRAKT20.TUT2PLI.LOAD(PLI02)'
```

Aufgabe: Schreiben Sie ein eigenes PLI-Programm, das "Tutorial 2, PLI-Version", Ihren Namen und Ihr S/390-Server-Login auf dem Bildschirm ausgibt. Falls Ihr Programm mehrere Autoren hat, müssen alle Namen dieser Autoren von Ihrem Programm ausgegeben werden. Speichern Sie Ihr Prorammm im Dataset PRAKT20.TUT2PLI.PLI02 als Version V2 ab.

Erstellen Sie ebenfalls ein neues JCL-Script PRAKT20.TUT2PLI.SCRIPTE(V2) mit angepassten Pfaden.

Erzeugen Sie einen per CALL ausführbaren Programmcode.

Führen Sie Ihr Programm aus.

Löschen Sie nichts, damit Ihr Betreuer sich alle Daten anschauen sowie Ihr Programm aufrufen kann.

Aufgabe: Erzeugen Sie per Tastendruck <ALT-Druck> einen Screenshot, der den Programmcode Ihres Programms enthält.

Erzeugen Sie ebenfalls einen Screenshot, der alle Ausgaben Ihres Programms enthält.

Alle Screenshots sind im JPG-Format zu erstellen. Ihre Größe darf 150 KByte nicht überschreiten. Im Ausnahmefall kann auch das Bitmap-Format verwendet werden. Ein solcher Screenshot darf nicht größer als 500 KByte werden.

Hängen Sie beide Screenshots als je einen Anhang an eine Mail (nicht in ein Zip-Archiv packen) und schicken Sie diese Ihrem Betreuer zu.