

z/OS Job Control Language

Übungen zur Schulung

6. Februar 2010

Eine Ausarbeitung von:

cps4it

Ralf Seidler • Stromberger Straße 36A • 55411 Bingen
Fon: +49-6721-992611 • Fax: -49-6721-992613 • Mail: ralf.seidler@cps4it.de

Internet : <http://www.cps4it.de>

Steuernummer: 08/220/2497/3, Finanzamt Bingen, Ust-ID : DE214792185

Diese Seite bleibt frei

Inhaltsverzeichnis

1. VORBEREITUNGEN	4
1.1 ANMELDEN UND TEST DER USER-IDENT	4
1.2 BIBLIOTHEK FÜR ÜBUNGSAUFGABEN ERSTELLEN	4
2. JOBKARTE	5
2.1 ERSTELLEN EINER JOBANWEISUNG	5
2.2 JOBKARTE ERWEITERN - 1	5
2.3 JOBKARTE ERWEITERN - 2	5
2.4 JOBSTEP ERSTELLEN	6
2.5 JOB MIT 2 STEPS	6
2.6 JOB MIT COND-STEUERUNG	6
3. ARBEITEN MIT DATEIEN – 1	7
3.1 LESEN EINER INSTREAM-DATEI	7
3.2 LESEN EINER DUMMY-DATEI	7
3.3 SCHREIBEN IN DAS „NIRWANA“	7
4. ARBEITEN MIT DATEIEN – 2	8
4.1 ANLEGEN EINER SEQUENTIELLEN DATEI	8
4.2 ANLEGEN EINER PO-DATEI	8
4.3 KOPIEREN IN EIN MEMBER EINER PO-DATEI	9
4.4 KOPIEREN PO-MEMBER	9
5. UTILITIES	10
6. JOBSTEUERUNG, STEPSTEUERUNG	11
6.1 ANLEGEN EINER SEQUENTIELLEN DATEI MIT RÜCKBEZUG	11
6.2 KOPIEREN EINER DATEI MIT EVENTUELLER NEUANLAGE	11
7. JOBS MIT LADEBIBLIOTHEKEN	12
7.1 PO-DATEI ALS LADEBIBLIOTHEK ANLEGEN	12
7.2 JOB MIT STEPLIB	12
7.3 JOB MIT JOBLIB	12
7.4 JOB MIT FALSCHEN PROGRAMMNAMEN	12
7.5 ANLEGEN GDG-BASE-ENTRY	13
7.6 KOPIEREN DATEN AUF GDS	13
7.7 ARBEITEN MIT AKTUELLEM GDS	13
7.8 ARBEITEN MIT ALLEN GENERATIONEN EINER GDG	13
7.9 LÖSCHEN GDG	13
7.10 STORCLAS IM UNTERNEHMEN	14
7.11 MGMTCLAS IM UNTERNEHMEN	14
8. JOBS MIT PROZEDUREN	15
8.1 PROZEDURAUFRUF - INSTREAM	15
8.2 PROZEDURAUFRUF – INSTREAM, PARAMETRISIERT	15
8.3 PROZEDURAUFRUF – EXTERN	15
8.4 PROZEDURAUFRUF – EXTERN, PARAMETRISIERT	15
8.5 PROZEDURAUFRUF – MSGLEVEL	15

1. Vorbereitungen

1.1 Anmelden und Test der User-Iden

Melden Sie sich nach Vorgabe im TSO an.
Melden Sie sich wieder ab und erneut wieder an.
Konfigurieren Sie Ihre TSO-Session, wie Sie am besten arbeiten können.
Testen Sie, ob Sie auf die Dateien des Referenten lesend zugreifen können.

1.2 Bibliothek für Übungsaufgaben erstellen

Erstellen Sie eine PO-Datei, um die Übungsaufgaben dieses Seminars aufnehmen zu können.

Name der Bibliothek	userid.KURS.JCL
Satzlänge	80 Byte
Satzformat	fest geblockt
Dateigröße	5 Spuren primär, 2 Spuren sekundär
Directory	Platz für 30 Member

Kontrollieren Sie das Ergebnis.

2. Jobkarte

2.1 Erstellen einer Jobanweisung

Member: \$JOB CARD

Kodieren Sie eine JOB Anweisung, eine Jobkarte, gemäß den Vorgaben in der Firma.

Welche Angaben sind verpflichtend?

Schreiben Sie eine minimale JOB Anweisung.

Welche Angaben sind frei wählbar?

Welche Angaben sind freiwillig?

Lassen Sie den Job laufen. Was geschieht?

2.2 Jobkarte erweitern - 1

Member: \$JOB CARD

Erweitern Sie die Jobkarte mit dem Parameter CLASS.

Erweitern Sie die Jobkarte mit dem Parameter MSGCLASS.

Lassen Sie den Job laufen und analysieren Sie die Ausgabe.

2.3 Jobkarte erweitern - 2

Member: \$JOB CAR2

Erweitern Sie die Jobkarte mit dem Parameter NOTIFY.

Lassen Sie den Job mit unterschiedlichen Angaben für NOTIFY laufen und analysieren Sie die Ausgabe.

Erweitern Sie die Jobkarte mit dem Parameter REGION.

Lassen Sie den Job mit unterschiedlichen Angaben für REGION laufen und analysieren Sie die Ausgabe.

Erweitern Sie die Jobkarte mit dem Parameter TIME.

Lassen Sie den Job mit unterschiedlichen Angaben für TIME laufen und analysieren Sie die Ausgabe.

2.4 Jobstep erstellen

Member: JCL201

Erweitern Sie die Jobkarte um den Parameter MSGLEVEL.
Fügen Sie einen Step ein mit dem Programm IEFBR14.

Lassen Sie den Job mit unterschiedlichen Angaben für MSGLEVEL laufen
und analysieren Sie die Ausgabe.

2.5 Job mit 2 Steps

Member: JCL202

Erweitern Sie den Job JCL201 um einen zweiten Step, der ebenfalls das Programm IEFBR14 aufruft.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe.
Was hat sich gegenüber der Ausgabe von JCL201 verändert?
Woran erkennen Sie, dass der 2. Step gelaufen ist?

2.6 Job mit COND-Steuerung

Member: JCL203:

Erweitern Sie den Job JCL202 so, dass der 2. Step bei der Ausführung des Jobs nicht angestoßen wird.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe.
Was hat sich gegenüber der Ausgabe von JCL202 verändert?
Woran erkennen Sie, dass der 2. Step nicht gelaufen ist?
Testen Sie den Job mit unterschiedlichen Angaben für die COND-Steuerung.

3. Arbeiten mit Dateien – 1

3.1 Lesen einer Instream-Datei

Member: JCL301

Erstellen Sie einen Job, der eine Instream-Datei einliest und auf die Spool ausgibt.

Nutzen Sie das Programm IEBGENER. Die DD-Anweisung für die Eingabe lautet SYSUT1, für die Ausgabe SYSUT2.
SYSIN DD DUMMY

Lassen Sie den Job laufen und prüfen Sie die Ausgabe.

3.2 Lesen einer Dummy-Datei

Member: JCL302

Erstellen Sie einen Job, der eine leere Datei, eine Dummy-Datei, liest und ausgibt.

Nutzen Sie das Programm IEBGENER.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe.

Woran erkennen Sie, dass das Programm richtig gelaufen ist?

3.3 Schreiben in das „Nirwana“

Member: JCL303

Erstellen Sie einen Job, der eine Instream-Datei einliest und die Ausgabe auf Dummy schreibt.

Nutzen Sie das Programm IEBGENER.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe.

Woran erkennen Sie, dass das Programm richtig gelaufen ist?

4. Arbeiten mit Dateien – 2

4.1 Anlegen einer sequentiellen Datei

Member: JCL401

Erstellen Sie einen Job, der eine sequentielle Datei userid.TEST.PS anlegt. Die Datei hat eine feste Länge von 80 Stellen und ist geblockt. Wählen Sie die Angaben so, dass mindestens 1000 Zeilen in diese Datei passen.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe. Woran erkennen Sie, dass die Datei richtig angelegt worden ist?

Gehen Sie anschließend im EDIT in die Datei und schreiben Sie einen beliebigen Inhalt hinein. Testen Sie, dass mindestens 1000 Zeilen, aber nicht mehr als 1500 Zeilen in der Datei Platz haben. Wie prüfen Sie die Grenze des Erreichbaren?

4.2 Anlegen einer PO-Datei

Member: JCL402

Erstellen Sie einen Job, der eine PO-Datei userid.TEST.PDS anlegt. Die Datei hat eine feste Länge von 80 Stellen und ist geblockt. Wählen Sie die Angaben so, dass mindestens 1000 Zeilen in die einzelnen Member passen und mindestens 20 Member in das Directory aufgenommen werden können.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe. Überprüfen Sie die Dateidefinition an Hand der Ausgabe des Jobs und auf einem anderen beliebigen Weg Ihrer Wahl.

4.3 Kopieren in ein Member einer PO-Datei

Member: JCL403

Erstellen Sie einen Job, der den von Ihnen erzeugten Inhalt der Datei aus Aufgabe 4.1 in das Member AUSSEQ der PO-Datei userid.TEST.PDS überträgt.

Lassen Sie den Job laufen und überprüfen Sie die Ausgabe und das Ergebnis.

4.4 Kopieren PO-Member

Member: JCL404

Erstellen Sie einen Job, der das Member AUSSEQ der PO-Datei userid.TEST.PDS in das Member AUSSEQ2 der gleichen PO-Datei kopiert.
Geht das überhaupt?
Kann das auch verhindert werden?

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.

5. Utilities

Keine Übungen bzw. siehe Unterlagen zu Kurs z/OS Dienstprogramme

6. Jobsteuerung, Stepsteuerung

6.1 Anlegen einer sequentiellen Datei mit Rückbezug

Member: JCL601

Erstellen Sie einen Job, dessen erster Step die sequentielle Datei userid.TEST.PS11 erzeugt. Die Datei hat eine feste Satzlänge von 80 Bytes und ist geblockt. Die Datei soll mindestens 1000 Zeilen aufnehmen können. Erstellen Sie einen weiteren Step, der die sequentielle Datei userid.TEST.PS12 erzeugt. Die Angaben für die Datei sollen über einen Rückbezug beschrieben werden.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.

Schreiben Sie im Edit einige Zeilen in die Datei userid.TEST.PS11.

6.2 Kopieren einer Datei mit eventueller Neuanlage

Member: JCL602

Erstellen Sie einen Job mit dem folgenden Ablauf:

- Anlegen sequentielle Datei userid.TEST.PS21 (Angaben wie oben)
- Falls Anlegen nok
 - Löschen Datei userid.TEST.PS21
 - Datei userid.TEST.PS21 neu anlegen (Angaben wie oben)
- Kopieren Inhalt von userid.TEST.PS11 nach userid.TEST.PS21

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.

7. Jobs mit Ladebibliotheken

7.1 PO-Datei als Ladebibliothek anlegen

Member: JCL701

Erstellen Sie einen Job der eine Ladebibliothek mit dem Namen userid.TEST.LOAD erstellt. Holen Sie sich die Parameter aus einer Systembibliothek.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.
Kann diese Datei auch über ISPF 3.2 erzeugt werden?
Kopieren Sie ein beliebiges Programm aus einer Systembibliothek hinein.

7.2 Job mit STEPLIB

Member: JCL702

Erstellen Sie einen Job, der auf ein / das Programm aus der Bibliothek userid.TEST.LOAD zugreift. Benutzen Sie eine STEPLIB-Anweisung.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.
Woran können Sie erkennen, dass Sie Ihre Bibliothek und nicht die Systembibliothek benutzt haben?

7.3 Job mit JOBLIB

Member: JCL703

Erstellen Sie einen Job, der auf ein / das Programm aus der Bibliothek userid.TEST.LOAD zugreift. Benutzen Sie eine JOBLIB-Anweisung.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.
Woran können Sie erkennen, dass Sie Ihre Bibliothek und nicht die Systembibliothek benutzt haben?

7.4 Job mit falschem Programmnamen

Member: JCL704

Erstellen Sie einen Job, der ein nicht vorhandenes Programm aufruft.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe.
Woran können Sie in der Jobausgabe den falschen Namen erkennen?

7.5 Anlegen GDG-Base-Entry

Member: JCL705

Erstellen Sie einen Job, der ein GDG-Base-Entry mit 3 Generationen erzeugt.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.

7.6 Kopieren Daten auf GDS

Member: JCL706

Erstellen Sie einen Job, der Daten auf eine neue Generation des GDG kopiert. Benutzen Sie als Eingabe die Datei userid.TEST.PS11.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis. Lassen Sie den Job mindestens 3 Mal laufen und prüfen Sie jeweils die Ergebnisse.

7.7 Arbeiten mit aktuellem GDS

Member: JCL707

Erstellen Sie einen Job, der die aktuelle Generation des GDG auf Spool ausgibt.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.

7.8 Arbeiten mit allen Generationen einer GDG

Member JCL708

Erstellen Sie einen Job, der alle Generationen des GDG auf Spool ausgibt.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.

7.9 Löschen GDG

Member: JCL709

Erstellen Sie einen Job, der die GDG mit allen GDS löscht.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis. (del dsn gdg recovery)

7.10 STORCLAS im Unternehmen

Prüfen Sie im ISPF, welche Storage-Klassen im Unternehmen existieren.

7.11 MGMTCLAS im Unternehmen

Prüfen Sie im ISPF, welche Management-Klassen im Unternehmen existieren.

8. Jobs mit Prozeduren

8.1 Prozeduraufruf - Instream

Member JCL801

Erstellen Sie einen Job auf der Basis von JCL401. Der Step von JCL401 soll als Instream-Prozedur aufgerufen werden.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.

8.2 Prozeduraufruf – Instream, parametrisiert

Member JCL802

Erstellen Sie einen Job, der die vorige Prozedur mit einem Parameter aufruft. Parameter soll der Dateiname sein.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.

8.3 Prozeduraufruf – extern

Member JCL803

Legen Sie die in der Aufgabe 8.1 definierte Prozedur in der Datei userid.KURS.JCL als Member PRC001 an. Erstellen Sie einen Job, der diese Prozedur aufruft.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.

8.4 Prozeduraufruf – extern, parametrisiert

Member: JCL804

Erstellen Sie einen Job analog der Aufgabe 8.2, der die Prozedur PRC002 aus der Datei userid.KURS.JCL aufruft. Bereiten Sie die Prozedur entsprechend vor.

Lassen Sie den Job laufen und prüfen Sie die Ausgabe und das Ergebnis.

8.5 Prozeduraufruf – MSGLEVEL

Testen Sie die Aufgaben 8.1, 8.2, 8.3 und 8.4 mit unterschiedlichen Angaben für MSGLEVEL: Prüfen Sie die Unterschiede der Ergebnisse.