

System Display and Search Facility (SDSF)

© Abteilung Technische Informatik, Institut für Informatik, Universität Leipzig
© Abteilung Technische Informatik, Wilhelm Schickard Institut für Informatik, Universität Tübingen

Falls Sie die Meldung MAXCC=0 (oder MAXCC=4) in Abb. 4.10 erhalten haben, könnten Sie eigentlich diesen Abschnitt überspringen. Sie werden die Information in Abschnitt 6 dennoch früher oder später brauchen, deswegen arbeiten Sie ihn bitte durch. Deshalb

Aufgabe: Erstellen Sie die am Ende von Abschnitt 6 gewünschten Screenshots

Die System Display and Search Facility (SDSF) ist eine Sammlung von Werkzeugen, welche bei der Fehlersuche behilflich sein können. Diese Sammlung ist sehr mächtig, aber nicht sehr intuitiv. Das Software Paket „Rational Developer for System z“ (RDz) bietet modernere Alternativen zu SDSF. Wir werden uns dies in einem späteren Tutorial ansehen.

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
VIEW          PRAK401.TEST.C(V1) - 01.00                      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.
000100 #include <stdio.h>
000200 main()
000300 {
000400     printf("Hallo Welt, unser erstes TSO-Programm \n");
000500 }
***** ***** Bottom of Data *****

Command ===>
F1=Help      F2=Split    F3=Exit     F5=Rfind    F6=Rchange  F7=Up
F8=Down     F9=Swap     F10=Left   F11=Right   F12=Cancel

Scroll ===> PAGE
```

Abb. 6.1

Angenommen, wir haben dieses C Programm,

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
VIEW          PRAK401.TEST.CNTL(V1) - 01.01                      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.
000100 //PRAK401C JOB ( ),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
000200 //          TIME=1440,REGION=0K
000300 //PROCLIB JCLLIB ORDER=CBC.SCCNPRC
000400 //CCL      EXEC PROC=EDCCB,
000500 //          INFILE='PRAK401.TEST.C(V1)',
000600 //          OUTFILE='PRAK401.TEST.LOAD(V1),DISP=SHR'
***** ***** Bottom of Data *****

Command ==> sub                      Scroll ==> PAGE
F1=Help      F2=Split      F3=Exit      F5=Rfind      F6=Rchange    F7=Up
F8=Down      F9=Swap        F10=Left     F11=Right     F12=Cancel

```

Abb. 6.2

und das dazugehörige JCL Script erzeugt. Wir geben das sub Kommando ein-

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
VIEW          PRAK401.TEST.CNTL(V1) - 01.01                      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.
000100 //PRAK401C JOB ( ),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
000200 //          TIME=1440,REGION=0K
000300 //PROCLIB  JCLLIB ORDER=CBC.SCCNPRC
000400 //CCL      EXEC PROC=EDCCB,
000500 //          INFILE='PRAK401.TEST.C(V1)',
000600 //          OUTFILE='PRAK401.TEST.LOAD(V1),DISP=SHR'
***** ***** Bottom of Data *****

IKJ56250I JOB PRAK401C(JOB09165) SUBMITTED
***
```

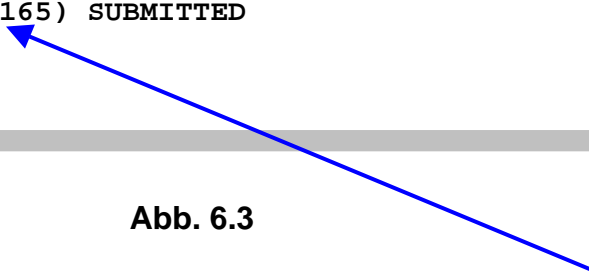


Abb. 6.3

JES hat gerade den Job akzeptiert, und wird ihn unter der Job Nummer 09165 ausführen. Diese Job Nummer notieren wir uns auf einem Stück Papier

```
09.18.07 JOB09165 $HASP165 PRAK401C ENDED AT N1 MAXCC=0 CN(INTERNAL)
***
```

Abb. 6.4

Es sei angenommen, alles ist ok, deshalb MAXCC= 0

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
VIEW          PRAK401.TEST.CNTL(V1) - 01.01                      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.
000100 //PRAK401C JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
000200 //          TIME=1440,REGION=0K
000300 //PROCLIB JCLLIB ORDER=CBC.SCCNPRC
000400 //CCL      EXEC PROC=EDCCB,
000500 //          INFILE='PRAK401.TEST.C(V1)',
000600 //          OUTFILE='PRAK401.TEST.LOAD(V1),DISP=SHR'
***** ***** Bottom of Data *****

Command ==>          Scroll ==> PAGE
F1=Help      F2=Split    F3=Exit      F5=Rfind     F6=Rchange   F7=Up
F8=Down      F9=Swap      F10=Left     F11=Right    F12=Cancel
```

Abb. 6.5

Enter bringt uns zurück.

Jetzt 4 mal die F3 Taste betätigen, um zum ISPF Primary Option Menu zurück zu gelangen.

```

Menu Utilities Compilers Options Status Help
-----
                    ISPF Primary Option Menu

0 Settings          Terminal and user parameters      User ID . : PRAK401
1 View             Display source data or listings      Time. . . : 09:11
2 Edit            Create or change source data        Terminal. : 3278
3 Utilities       Perform utility functions          Screen. . : 1
4 Foreground     Interactive language processing          Language. : ENGLISH
5 Batch          Submit job for language processing          Appl ID . : ISR
6 Command        Enter TSO or Workstation commands          TSO logon : DBSPROC
7 Dialog Test    Perform dialog testing                          TSO prefix: PRAK401
9 IBM Products   IBM program development products                System ID : ADCD
                                                    MVS acct. : ACCT
                                                    Release . : ISPF 5.8
+-----+-----+-----+-----+-----+-----+ r
| Licensed Materials - Property of IBM                |
| 5694-A01 (C) Copyright IBM Corp. 1980, 2006.      |
| All rights reserved.                               |
| US Government Users Restricted Rights -            |
| Use, duplication or disclosure restricted          | s
| by GSA ADP Schedule Contract with IBM Corp.       |
+-----+-----+-----+-----+-----+-----+
Option ==>
  F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
  F10=Actions  F12=Cancel

```

Abb. 6.6

Mittels der F8 Taste entfernen wir die uninteressante Meldung „Licensed Materials usw.“,

```

Menu Utilities Compilers Options Status Help
-----
                    ISPF Primary Option Menu

0 Settings      Terminal and user parameters      User ID . . : PRAK401
1 View         Display source data or listings          Time. . . . : 09:21
2 Edit        Create or change source data      Terminal. . : 3278
3 Utilities    Perform utility functions                Screen. . . : 1
4 Foreground  Interactive language processing          Language. . : ENGLISH
5 Batch       Submit job for language processing      Appl ID . . : ISR
6 Command     Enter TSO or Workstation commands            TSO logon  : DBSPROC
7 Dialog Test Perform dialog testing                TSO prefix: PRAK401
9 IBM Products IBM program development products    System ID  : ADCD
10 SCLM       SW Configuration Library Manager           MVS acct.  : ACCT
11 Workplace  ISPF Object/Action Workplace                Release . . : ISPF 5.8
M More        Additional IBM Products

Enter X to Terminate using log/list defaults

Option ==> M
  F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
 F10=Actions  F12=Cancel

```

Abb. 6.7

und geben M auf der Command Line ein. Damit blättern wir zum Folgescreen (aus Platzgründen braucht das ISPF Primary Option Menue zwei Panels).

(Alternativ, Eingabe des TSO SDSF Commands von der Command line ==> ruft ebenfalls SDSF auf.)

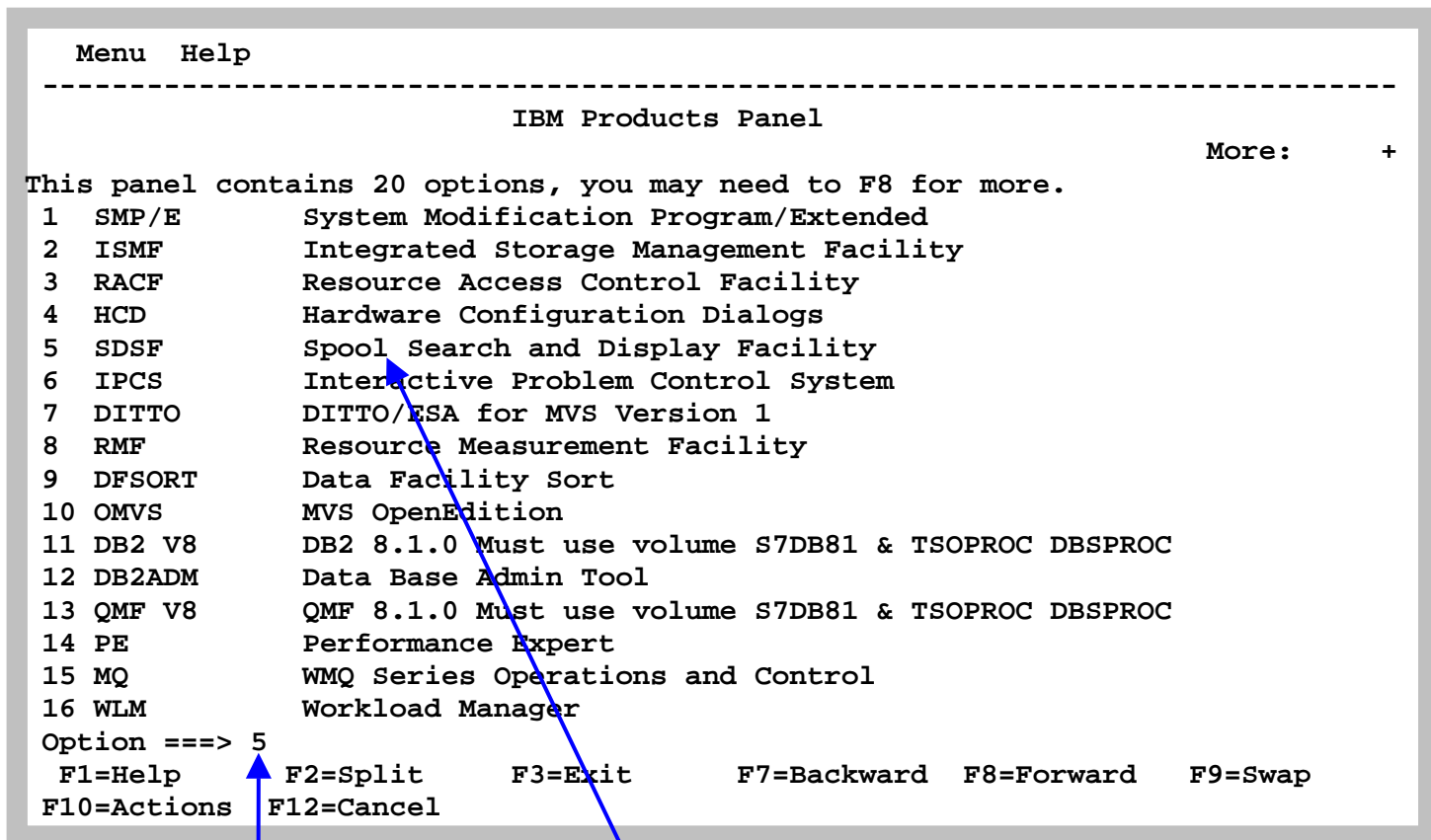


Abb. 6.8

Toll, was es unter ISPF da so alles an Möglichkeiten gibt. 3 RACF benutzen Sie z.B als Systemadministrator, um Sicherheitseinstellungen zu verwalten, oder eine neue User ID anzulegen. 16 WLM (Work Load Manager) ist eine hervorragende Einrichtung, um das Leistungsverhalten eines Rechners zu optimieren. Wir werden uns WLM später anschauen. Hier geben wir eine „5“ ein, um die „System Display and Search“ (SDSF) Facility aufzurufen.

JES speichert eine Reihe von Zwischenergebnissen in temporären Files ab. Ganz früher wurden diese Data Sets auf Magnetband geschrieben (Spool) um dann bei Bedarf offline ausgedruckt zu werden. Deshalb werden diese Files als Spool Files bezeichnet.

Die Spool Files enthalten auch Status Information über den fortschritt bei der Ausführung unseres Jobs. Das kann vor allem im Fehlerfall interessant sein.

Wir geben eine 5 ein, um das SDSF Subsystem aufzurufen, und ...

gelangen in das SDSF Primary Option Menu, einem ISPF Subsystem. Es bedeuten:

- DA Display Active zur Überwachung der Jobs die im System ausgeführt werden.
- I Input Queue zur Überwachung der Jobs, die zur Ausführung bereit stehen.
- H Held output queue, um die Ergebnisse der Job-Ausführung zu überprüfen

```
Display Filter View Print Options Help
-----
HQX7730 ----- SDSF PRIMARY OPTION MENU -----
-----
DA Active users
I Input queue
O Output queue
H Held output queue
ST Status of jobs

SE Scheduling environments

END Exit SDSF

Licensed Materials - Property of IBM

5694-A01 (C) Copyright IBM Corp. 1981, 2006. All rights reserved.
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

COMMAND INPUT ==> ST SCROLL ==> PAGE
F1=HELP F2=SLIT F3=END F4=RETURN F5=IFIND F6=BOOK
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE
```

Abb. 6.9

Wir geben ST auf der Kommandozeile ein, um uns nach dem Status unseres Jobs zu erkundigen.

Die ersten beiden Spalten werden als NP (Non Protected) bezeichnet. In diese Spalten kann man Kommandos eingeben, vor allem die „?“ und „S“ Kommandos.

```

Display  Filter  View  Print  Options  Help
-----
SDSF STATUS DISPLAY ALL CLASSES                               LINE 1-2 (2)
NP  JOBNAME  JobID   Owner   Prty Queue      C  Pos  SAff  ASys Status
   PRAK401  TSU09164 PRAK401  15 EXECUTION      SYS1  SYS1
?  PRAK401C JOB09165 PRAK401   1 PRINT          A   402

COMMAND INPUT ==>
F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE

SCROLL ==> PAGE

```

Abb. 6.10

In diesem Beispiel ist der Screen ungewöhnlich leer. Meistens befinden sich hier zahlreiche Einträge für unterschiedliche Jobs. Aber wir finden hier den Eintrag für unseren Job 09165. Wir geben ein Fragezeichen (?) in diese Zeile in der NP Spalte ein, um nachzuschauen.

Dies bewirkt, dass die bei der Ausführung unseres Jobs angelegten Spool Files angezeigt werden.

Display Filter View Print Options Help										
SDSF JOB DATA SET DISPLAY - JOB PRAK401C (JOB09165)										LINE 1-4 (4)
NP	DDNAME	StepName	ProcStep	DSID	Owner	C	Dest	Rec-Cnt	Page	
S	JESMSG LG	JES2		2	PRAK401	H	LOCAL	14		
	JESJCL	JES2		3	PRAK401	H	LOCAL	111		
	JESYSMSG	JES2		4	PRAK401	H	LOCAL	119		
	SYSPRINT	CCL	BIND	105	PRAK401	H	LOCAL	189		

COMMAND INPUT	====>							SCROLL	====>	PAGE
F1=HELP		F2=SPLIT		F3=END		F4=RETURN		F5=IFIND		F6=BOOK
F7=UP		F8=DOWN		F9=SWAP		F10=LEFT		F11=RIGHT		F12=RETRIEVE

Abb. 6.11

Wiedergegeben werden drei Data Set Namen (dd names): JES2 messages log file, JES2 JCL file, und JES2 system Messages File unseres Jobs 09165.

JES notiert alle Aktionen für alle Jobs in einer Log File JESMSG LG. Ein S in der NP Spalte zeigt jeweils den Inhalt einer der Files an.

```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY PRAK401C JOB09165  DSID      2 LINE 0          COLUMNS 02- 81
COMMAND INPUT ==>                               SCROLL ==> PAGE
***** TOP OF DATA *****
                J E S 2  J O B  L O G  --  S Y S T E M  S Y S 1  --  N

09.18.05 JOB09165  ---- WEDNESDAY, 11 JUL 2012 ----
09.18.05 JOB09165  IRR010I  USERID PRAK401  IS ASSIGNED TO THIS JOB.
09.18.05 JOB09165  ICH70001I PRAK401  LAST ACCESS AT 09:11:49 ON WEDNESDAY, JULY
09.18.05 JOB09165  $HASP373 PRAK401C STARTED - INIT 1      - CLASS A - SYS SYS1
09.18.06 JOB09165  $HASP395 PRAK401C ENDED
----- JES2 JOB STATISTICS -----
  11 JUL 2012 JOB EXECUTION DATE
      6 CARDS READ
     433 SYSOUT PRINT RECORDS
       0 SYSOUT PUNCH RECORDS
      20 SYSOUT SPOOL KBYTES
     0.01 MINUTES EXECUTION TIME
  1 //PRAK401C JOB ( ),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
    //                TIME=1440,REGION=0K
    IEF653I SUBSTITUTION JCL - ( ),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIF
F1=HELP      F2=SPLIT      F3=END        F4=RETURN      F5=IFIND      F6=BOOK
F7=UP        F8=DOWN       F9=SWAP      F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Abb. 6.12

Dies ist der erste von vielen Screens, die man sich mit F8 und F7 blättern und ansehen kann.

Mit F3 Zurück zum vorherigen Panel.

```

Display  Filter  View  Print  Options  Help
-----
SDSF JOB DATA SET DISPLAY - JOB PRAK401C (JOB09165)      DATA SET DISPLAYED
NP  DDNAME  StepName ProcStep DSID Owner  C Dest  Rec-Cnt Page
   JESMSGLG JES2          2 PRAK401 H LOCAL  14
   JESJCL   JES2          3 PRAK401 H LOCAL 111
S   JESYSMSG JES2          4 PRAK401 H LOCAL 119
   SYSPRINT CCL          BIND    105 PRAK401 H LOCAL 189

COMMAND INPUT ==>
F1=HELP      F2=SPLIT    F3=END      F4=RETURN   F5=IFIND    F6=BOOK
F7=UP        F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE

SCROLL ==> PAGE

```

Abb. 6.13

Schauen wir uns den Inhalt von JESYSMSG an, in dem wir ein „S“ eingeben.

```

  Display  Filter  View  Print  Options  Help
-----
SDSF OUTPUT DISPLAY PRAK401C JOB09165  DSID      4 LINE 0      COLUMNS 02- 81
COMMAND INPUT ==>                                SCROLL ==> PAGE
***** TOP OF DATA *****
STMT NO. MESSAGE
      3 IEF001I PROCEDURE EDCCB WAS EXPANDED USING PRIVATE LIBRARY CBC.SCCNPR
ICH70001I PRAK401  LAST ACCESS AT 09:11:49 ON WEDNESDAY, JULY 11, 2012
IEF236I ALLOC. FOR PRAK401C COMPILE CCL
IEF237I 6021 ALLOCATED TO STEPLIB
IEF237I 6021 ALLOCATED TO
IEF237I 6021 ALLOCATED TO
IEF237I 601B ALLOCATED TO SYSIN
IEF237I 6021 ALLOCATED TO SYSLIB
IEF237I 6021 ALLOCATED TO
IGD100I 6011 ALLOCATED TO DDNAME SYSLIN  DATACLAS (      )
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO SYSOUT
IEF237I JES2 ALLOCATED TO SYSCPRT
IGD100I 6013 ALLOCATED TO DDNAME SYSUT5  DATACLAS (      )
IGD100I 6011 ALLOCATED TO DDNAME SYSUT6  DATACLAS (      )
IGD100I 601B ALLOCATED TO DDNAME SYSUT7  DATACLAS (      )
  F1=HELP      F2=SPLIT      F3=END        F4=RETURN     F5=IFIND      F6=BOOK
  F7=UP        F8=DOWN       F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE

```

Abb. 6.14

Das JESYSMSG Listing enthält Memory Allocation und Cleanup Messages. ALLOC sagt, welche Devices und wieviel Hauptspeicherplatz für den Job Step allocated wurde. Sie erinnern sich, ein Job Step führt ein Programm aus. Es informiert ebenfalls über die von dem Job Step verbrauchte CPU Zeit.

Mit F3 zurück zum SDSF Status Display all Classes Panel.

```

Display  Filter  View  Print  Options  Help
-----
SDSF STATUS DISPLAY ALL CLASSES                                LINE 1-2 (2)
NP  JOBNAME  JobID   Owner   Prty Queue      C  Pos  SAff  ASys Status
   PRAK401  TSU09180 PRAK401   15 EXECUTION      SYS1  SYS1
S  PRAK401C  JOB09165 PRAK401    1 PRINT          A   402

COMMAND INPUT ==>
F1=HELP      F2=SPLIT      F3=END        F4=RETURN      F5=IFIND      F6=BOOK
F7=UP        F8=DOWN       F9=SWAP       F10=LEFT       F11=RIGHT     F12=RETRIEVE
SCROLL ==> PAGE

```

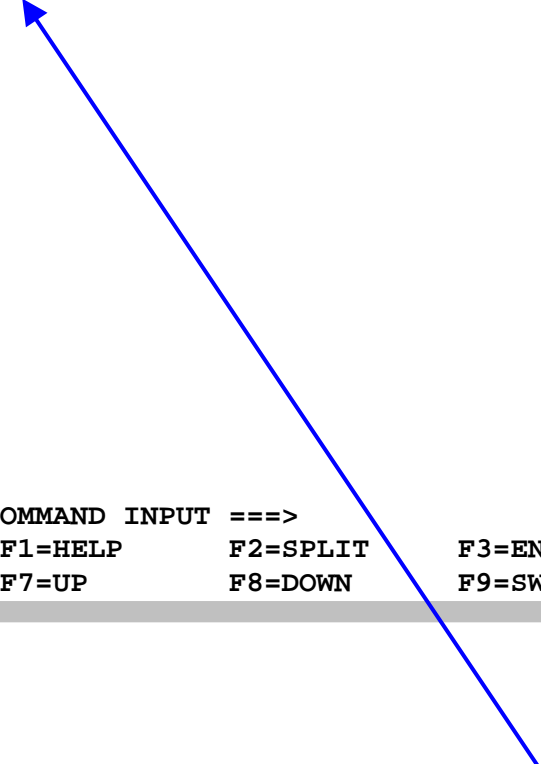


Abb. 6.15

Um eine Zusammenfassung aller Log Files zu sehen, die bei der Ausführung von Job 09165 erzeugt wurden, geben wir ein "S" an Stelle des Fragezeichens (?) in der NP Spalte ein.


```

Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY PRAK401C JOB09165  DSID      2 LINE 0          COLUMNS 02- 81
COMMAND INPUT ==>                               SCROLL ==> PAGE
***** TOP OF DATA *****
                J E S 2  J O B  L O G  --  S Y S T E M  S Y S 1  --  N

09.18.05 JOB09165  ---- WEDNESDAY, 11 JUL 2012 ----
09.18.05 JOB09165  IRR010I  USERID PRAK401  IS ASSIGNED TO THIS JOB.
09.18.05 JOB09165  ICH70001I PRAK401  LAST ACCESS AT 09:11:49 ON WEDNESDAY, JULY
09.18.05 JOB09165  $HASP373 PRAK401C STARTED - INIT 1      - CLASS A - SYS SYS1
09.18.06 JOB09165  $HASP395 PRAK401C ENDED
----- JES2 JOB STATISTICS -----
  11 JUL 2012 JOB EXECUTION DATE
      6 CARDS READ
     433 SYSOUT PRINT RECORDS
       0 SYSOUT PUNCH RECORDS
      20 SYSOUT SPOOL KBYTES
     0.01 MINUTES EXECUTION TIME
  1 //PRAK401C JOB ( ),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
    //              TIME=1440,REGION=0K
    IEF653I SUBSTITUTION JCL - ( ),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIF
F1=HELP      F2=SPLIT      F3=END        F4=RETURN      F5=IFIND      F6=BOOK
F7=UP        F8=DOWN       F9=SWAP      F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Abb. 6.16

Dies gibt eine Zusammenfassung der Files von Job09165 wieder. Der Screen Inhalt ist mit Abb. 6.12 identisch.

Dies ist der erste von vielen Screens, die man sich mit F8 und F7 blättern und ansehen kann. Interessant sind die letzten Screens

```
Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY PRAK401C JOB09165  DSID   105 LINE 152      COLUMNS 02- 81
COMMAND INPUT ==>                                SCROLL ==> PAGE
ENTRY POINT AND ALIAS SUMMARY:

NAME:          ENTRY TYPE AMODE C_OFFSET CLASS NAME          STATUS
CEESTART       MAIN_EP      31 00000000 B_TEXT

                ***** E N D   O F   R E P O R T *****

z/OS V1 R8 BINDER      09:18:05 WEDNESDAY JULY 11, 2012
BATCH EMULATOR JOB(PRAK401C) STEP(CCL      ) PGM= IEWL      PROCEDURE(BIND      )
IEW2008I 0F03 PROCESSING COMPLETED.  RETURN CODE =  0.

-----
F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=IFIND      F6=BOOK
F7=UP        F8=DOWN      F9=SWAP      F10=LEFT      F11=RIGHT     F12=RETRIEVE
```

Abb. 6.17

Hier wird bestätigt, dass der Return Code (MAXCC) den Wert 0 hat.

```
Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY PRAK401C JOB09165  DSID   105 LINE 170      COLUMNS 02- 81
COMMAND INPUT ==>                                           SCROLL ==> PAGE
MESSAGE SUMMARY REPORT
-----
TERMINAL MESSAGES      (SEVERITY = 16)
NONE

SEVERE MESSAGES        (SEVERITY = 12)
NONE

ERROR MESSAGES         (SEVERITY = 08)
NONE

WARNING MESSAGES       (SEVERITY = 04)
NONE

INFORMATIONAL MESSAGES (SEVERITY = 00)
2008  2278

F1=HELP      F2=SPLIT    F3=END      F4=RETURN   F5=IFIND    F6=BOOK
F7=UP        F8=DOWN      F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

Abb. 6.18

Das darauf folgende Panel beagt, die Fehler Messages sind alle ok

Im Fehlerfall würden die hier wiedergegebenen Files Tips und Anmerkungen zu dem aufgetretenen Fehler enthalten. Je nach Fehlerursache kann das sehr unterschiedlich sein.

Aufgabe: Bauen Sie in Ihr Cobol Programm oder in ihr JCL Script einen Fehler ein, und zeigen Sie in einem Screenshot wie SDSF hierfür Hilfe leistet.

Literatur zum Thema SDSF

ABCs of z/OS System Programming Volume 1, Abschnitt 4.5.1, S. 173 ff.
<http://www.informatik.uni-leipzig.de/cs/Literature/Textbooks/ispf/ABCs01.pdf>

Chip Wood: SDSF for New Users.
<http://www.cedix.de/VorlesMirror/Band3/sdsf03.pdf>

Ebenfalls Information über SDSF

<http://www.cedix.de/VorlesMirror/Band3/sdsf01.pdf>

<http://www.cedix.de/VorlesMirror/Band3/sdsf02.pdf>

Anhang A : Beispiel für ein weiteres Cobol Programm

Source Code

```
000100 ID DIVISION.
000200 PROGRAM-ID.  ACCEPT1.
000300 DATA DIVISION.
000400 WORKING-STORAGE SECTION.
000500 01  WS-FIRST-NUMBER      PIC 9(3).
000600 01  WS-SECOND-NUMBER     PIC 9(3).
000700 01  WS-TOTAL             PIC ZZZ9.
000800*
000900 PROCEDURE DIVISION.
001000 0000-MAINLINE.
001100     DISPLAY 'ENTER A NUMBER: '.
001200     ACCEPT WS-FIRST-NUMBER.
001300*
001400     DISPLAY 'ANOTHER NUMBER: '.
001500     ACCEPT WS-SECOND-NUMBER.
001600*
001700     COMPUTE WS-TOTAL = WS-FIRST-NUMBER + WS-SECOND-
NUMBER.
001800     DISPLAY 'THE TOTAL IS: ', WS-TOTAL.
001900     STOP RUN.
```

Sample Run

```
ENTER A NUMBER: 7
ANOTHER NUMBER: 7
THE TOTAL IS: 14
```

Abb. A.1 Beispiel für ein weiteres Cobol Programm

Anhang B: Cobol Fixed Format -Spaltenabhängigkeit

In fixed format, the COBOL source record is divided into the following areas:

Columns 1 - 6	Sequence number
Column 7	Indicator area
Column 8 - 11	Area A
Columns 12 - 72	Area B

COBOL source record can extend up to a maximum of 80 columns in length. The contents of columns 73 to 80 are ignored by the COBOL system.

Sequence Number

A sequence number of six digits can be used to identify each source program line.

If the first character position of the sequence number field contains an asterisk, or any nonprinting control character (less than the character SPACE in the ASCII collating sequence), then the line is treated as comment and is not output to the listing file or device. This facility allows an output listing file to be used as a source file to a subsequent compilation.

Indicator Area

An asterisk (*) in the indicator area marks the line as documentary comment only. Such a comment line can appear anywhere in the program after the IDENTIFICATION DIVISION header. Any characters from the ASCII character set can be included in area A and area B of the line.

A comment line can appear before the IDENTIFICATION DIVISION header.

A "D" or "d" in the indicator area represents a debugging line. Areas A and B can contain any valid COBOL sentence.

A "-" in the indicator area represents a continuation of the previous line without spaces or the continuation of a nonnumeric literal (see the section Continuation of Lines in the chapter Language Fundamentals).

A "\$" in the indicator area indicates a special line for setting directives or conditional compilation.

Areas A and B

Section names and paragraph names begin in area A and are followed by a period and a space.

Anhang C: Frequently asked Questions

What happens if you:

1. capitalize IDENTIFICATION DIVISION? **Nothing... cobol is not case sensitive**
 2. delete the period after identification division? **Compiler error... line2: Warning: '.' expected, PROGRAM-ID found**
 3. put an * in column 7 on the first statement (identification division)? **Nothing... weird, huh?!**
 4. change the program-id from prelab to sample? **Nothing... Remember that the program-id does not have to match the file name. I'll point out in a later question where this is used.**
 5. put the program-id name, prelab. (don't forget to include the period) on the next line starting in column 10? **Nothing...**
 6. put the program-id name, prelab. (don't forget to include the period) on the next line starting in column 12? **Nothing...**
 7. move data division to start in column 11? **Nothing...**
 8. move data division to start in column 12? **Compiler error... line12: Warning: Must start in Area A**
 9. delete the period on the line before the data division i.e. after the second organization is line sequential? **Compiler error... line 12: warning: Area A should be blank, line 12: numeric literal expected, DIVISION found, line 14: syntax error**
 10. comment out the 3 statements environment division, input-output section and file-control by putting an * in column 7 of each statement? **Nothing... weird, huh?!**
 11. comment out the file section statement? **6 compiler errors with the 7th one saying "Can't recover, good bye!"**
- NOTE:** Delete the * on the accept omitted statement (before the stop run statement) for the next question only.
12. in the file section, change the pic x(20) on 05 employee-name-in to pic x(18)? **Run error occurs 5 times... non-numeric data in numeric field at 000083 in PRELAB. The 000083 is a memory location which will probably change each time the program is run. The value is a hex value (binary = base 2, octal = base 8, decimal = base 10, hexadecimal = base 16). The PRELAB is the program-id variable name! The change also causes not just a run error, but a logic error as well! Notice that the output file has incorrect data...**
 13. change the 77 level number in the working-storage section to an 01 level number? **nothing**
 14. change the non-numeric literal on the value clause for the are-there-more-records variable to "no"? **a logic error occurs where in the output file, the page and column headers as well as the underline appear, but no data**

15. in the file section, put a \$ in front of the 9.99 on the pic clause for rate-out (i.e. pic \$9.99)? a \$ appears on the data under the \$/hr column; notice the earnings data was moved over one position
16. delete the period on the weekly-wages-out pic clause (i.e. pic 99999 instead of 999.99)? in the output file, leading zeroes and no decimal values appear on the data
17. delete the period before the working-storage section statement? **Compiler errors... line 30: Warning: Area A should be blank; line 30: syntax error scanning WORKING-STORAGE; line 31: illegal level number**
18. in the file section, change the variable name hours-worked-in to hrs-wrk-in? **compiler error - line 65: undefined data item: HOURS-WORKED-IN. Notice error on use of variable name, not declaration.**
19. in the file section, change the variable name weekly-wages-out to wkly-wgs-out? **Compiler error - Line 69: same as above with WEEKLY-WAGES-OUT**
20. change the variable name employee-data on the read statement to employee-record? **Compiler error - Line 55 or 71: no SELECT for file: EMPLOYEE-RECORD**
21. change the variable name hours-out in 200-wage-routine paragraph to hrs-out? **Compiler error – line 66: undefined data item: HRS-OUT**
22. change 05 levels on employee-record in the file section to 01 levels? **Compiler error – line 17: warning: must start in Area A, line 17: warning: no space in item: EMPLOYEE –RECORD i.e. no pic clause; same two errors for line 18**
23. change the double quotes to single quotes on the select statement? **nothing**
24. change the double quotes to single quotes for all the “no” non-numeric literals? **nothing**
25. change “no” to “NO” on the perform statement (in paragraph 100-main-module)? **Run error... box hard to see... press enter, maximize the window, and re-run the program. File error 46 on prelabin.dat. COBOL error at 00008C in prelab <OK>. Also, notice the output file has extra/duplicate last data line. Why is this an error?! Because non-numeric literal are case sensitive!!! i.e. “no” is not the same as “NO”.**
26. comment out the 100-main-module paragraph name? **compiler error line 46: identifier expected, OPEN found; line 46: missing PARA/SECTION; line 46: Can't recover, good-bye!**
27. comment out the 1st line in the 100-main-module paragraph (open input employee-data)? **Compiler error – line 47: verb expected, OUTPUT found. What if comment out the entire OPEN statement i.e. open line and next line as well?! You get a run error... box says file error 48.02 on prelabout.dat. COBOL error at 000010 in prelab <OK>.**
28. delete all the periods under the 100-main-module paragraph? **Compiler error – line 62: warning: missing period.**
29. delete all the periods under the 100-main-module paragraph except on the stop run statement? **Logic error! Output file has up to the first data line under page and column headers with underline.**

30. correct the change you made to the above question and forget to save before recompiling? **You get the same error since prelab.cob was not actually updated; just the screen version was updated, not the stored version.**
31. change the line before the read statement to after advancing 0 lines (instead of 1 lines)? **It looks like the column headers are deleted. Actually, this underlines the column headers but since the screen cannot put two characters in one location/space, the dashes overwrote the column header info. If you print the output, you will see the underlined column headers without advancing the print head one line (to be discussed in more detail when talk about the WRITE statement).**
32. change the line before the read statement to after advancing 1 line (deleting the s off the word lines)? **nothing**
33. comment out the close statement? **nothing**
34. put employee-data on the close statement on a different line, starting column 12 or after? **nothing**
35. comment out the stop run statement? Same as #27! Why?! **Because without a stop run, the program continues to run statements in sequential order including over the paragraph name and to the first statement in the paragraph, etc. and you can't write to a file that you have already closed!**

Other:

36. Notice that the 3rd line and the 11th line are blank; there is no * in the 7th column to denote a comment. Why is this okay?! **COBOL ignores "white space"!**
37. Notice that the non-numeric literal "Employee Name" has a length of 13, but the pic clause has a length of 23. Why is this okay? **Because the data is stored to the left so allows for open positions to the right.**
38. Notice that there are no variable names on the 05 levels under both page-header and column-headers. Why is this okay? **Because I don't need to access those memory locations**
39. List the first 20 statements executed by the program. Open, write, write, write, read, perform, move, move, move, multiply, write, read, perform, move, move, move, multiply, write
40. Delete the * on the accept omitted statement. Add the following display statement between the first read statement and the perform statement. Also add the following display statement at the end of the 2nd paragraph (i.e. the last sequential statement for the program). What output do you see on the screen? **Employee record is (data line information for each record) except the last (i.e.5th) record is repeated, so there are 6 output lines to the screen**
- What happens if you delete the period after the second read statement? **Only two lines of output appear; one for the first record, one for the last record.**

<http://www.cse.ohio-state.edu/~reeves/CSE314/Labs/Lab1PartBHWSolutions.htm>

see also <http://www.cse.ohio-state.edu/~reeves/CSE314/Labs/>

Anhang D : Coding JCL Statements

Identifier field

The identifier field indicates to the system that a statement is a JCL statement rather than data. The identifier field consists of the following:

- Columns 1 and 2 of all JCL statements, except the delimiter statement, contain // v Columns 1 and 2 of the delimiter statement contain either /* or two other characters designated in a DLM parameter to be the delimiter
- Columns 1, 2, and 3 of a JCL comment statement contain /**

Name field

The name field identifies a particular statement so that other statements and the system can refer to it. For JCL statements, code the name as follows:

- The name must begin in column 3. v The name is 1 through 8 alphanumeric or national (\$, #, @) characters. See Table 4-2 on page 4-3 for the character sets.
- The first character must be an alphabetic or national (\$, #, @).
- The name must be followed by at least one blank.

Operation field

The operation field specifies the type of statement, or, for the command statement, the command. Code the operation field as follows:

- The operation field consists of the characters in the syntax box for the statement. v The operation follows the name field.
- The operation must be preceded and followed by at least one blank.

Parameter, or operand field

The parameter field, also sometimes referred to as the operand field, contains parameters separated by commas. Code the parameter field as follows:

- The parameter field follows the operation field.
- The parameter field must be preceded by at least one blank.

Comments field

The comments field contains any information you deem helpful when you code the control statement. Code the comments field as follows:

- The comments field follows the parameter field.
- The comments field must be preceded by at least one blank.

Code the identifier field beginning in column 1 and the name field immediately after the identifier, with no intervening blanks. Code the operation, parameter, and comments fields in free form. Free form means that the fields need not begin in a particular column. Between fields leave at least one blank; the blank serves as the delimiter between fields.

Exceeding 71 columns

Do not code fields, except on the comment statement, past column 71. If the total length of the fields would exceed 71 columns, continue the fields onto one or more following statements

Continuing the Parameter Field

- 1. Interrupt the field after a complete parameter or subparameter, including the comma that follows it, at or before column 71.**
- 2. Code // in columns 1 and 2 of the following statement.**
- 3. Code a blank character in column 3 of the following statement. If column 3 contains anything but a blank or an asterisk, the system assumes the following statement is a new statement. The system issues an error message indicating that no continuation is found and fails the job.**
- 4. Continue the interrupted parameter or field beginning in any column from 4 through 16.**

**z/OSMVS JCL Reference SA22-7597-10 Eleventh Edition, April 2006 , chapter 3,
<http://www-01.ibm.com/support/docview.wss?uid=pub1sa22759712>**

Anhang E : Cobol Key Words

#+Here is a list of COBOL keywords that can start a line and end with the . character:

ACCEPT
ACQUIRE
ADD
ALTER
CALL
CANCEL
CLOSE
COMMIT
COMPUTE
DELETE
DISPLAY
DIVIDE
DROP
EXIT
EXIT PROGRAM
GO TO
GOBACK
INITIALIZE
INSPECT
MERGE
MOVE
MULTIPLY
OPEN
PERFORM
READ
RELEASE
RETURN
REWRITE
ROLLBACK
SET
SORT
START
STOP
STOP RUN
STRING
SUBTRACT
UNSTRING
WRITE

Some of these statements also have an implicit terminator:

**END-ACCEPT
END-ADD
END-CALL
END-COMPUTE
END-DELETE
END-DIVIDE
END-EVALUATE
END-IF
END-MULTIPLY
END-PERFORM
END-READ
END-RETURN
END-REWRITE
END-SEARCH
END-START
END-STRING
END-SUBTRACT
END-UNSTRING
END-WRITE**