

Übung Nr. 5

Nachdem wir eine DB2 Datenbank erfolgreich angelegt haben, wollen wir nun eine CICS Transaktion schreiben, die einen Zugriff auf diese Datenbank ausführt.

Die Vorgehensweise ist ähnlich wie in Übung 3. Unser Anwendungsprogramm soll wieder aus zwei Teilen bestehen, einem C-Programm für die Business Logik und einem BMS Programm für die Presentation Logik.

Der Unterschied zu Übung 3 besteht darin, daß unser Business Logik Programm SQL Aufrufe enthält. Diese müssen durch einen SQL Precompiler in die native API des DB2 Datenbanksystems übersetzt werden, ehe der C-Compiler das Business Logik Programm übersetzen kann.

Wir loggen uns als TSO Benutzer ein, und arbeiten uns zum DLIST Panel vor.

```
Menu  Options  View  Utilities  Compilers  Help
-----
DSLISL - Data Sets Matching SPRUTH                                     Row 1 of 13
Command - Enter "/" to select action                                Message                                Volume
-----
      SPRUTH                                                         *ALIAS
      SPRUTH.CICS.TEST04                                           SMS001
      SPRUTH.CICSDB2.TEST01                                        SMS001
      SPRUTH.DBRMLIB.DATA                                         SMS001
      SPRUTH.ISPF.ISPPROF                                         SMS001
      SPRUTH.LIB                                                  SMS001
      SPRUTH.SPFLOG1.LIST                                          SMS001
      SPRUTH.SPUFI.IN                                             SMS001
      SPRUTH.SPUFI.OUT                                            SMS001
      SPRUTH.TEST.C                                               SMS001
      SPRUTH.TEST.CNTL                                           SMS001
      SPRUTH.TEST.LOAD                                            SMS001
***** End of Data Set list *****
Command ==>
      F1=Help      F3=Exit      F5=Rfind  F12=Cancel
      Scroll ==> PAGE
```

Wir hatten bereits im Vorgriff alle hierfür erforderlichen Partitioned Data Sets angelegt. Unser DSLIST Panel zeigt 11 Partitioned Data Sets. SPUFI.IN wurde in unserer letzten Übung (Tutor4) benutzt, um unsere Datenbank anzulegen. SPUFI.OUT wurde vom SPUFI Subsystem angelegt und enthält die Übersetzung unserer Eingaben.

Im Vorgriff auf das Schreiben der CICS-DB2 Transaktion hatten wir noch die Data Sets SPRUTH.DBRMLIB.DATA und SPRUTH.CICSDB2.TEST01 angelegt. SPRUTH.DBRMLIB.DATA ist noch leer. Es wird während der Ausführung des SQL Precompiler automatisch gefüllt. SPRUTH.LIB wurde bei der Erstellung unserer letzten Transaktion SP04 (AbschnittTutor03) bei der Generierung des Mapsets mit einem Member versehen, der eine C-Struktur für die Ein/Ausgabe enthält. Ein weiterer Member kommt dazu wenn wir den Mapset für unsere neue Transaktion erstellen. SPRUTH.CICSDB2.TEST01 schließlich nimmt die von uns zu erstellenden Quellprogramme auf.


```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          SPRUTH.CICSDB2.TEST01(PRAKSET) - 01.00          Columns 00001 00072
***** ***** Top of Data *****
==MSG> -CAUTION- Profile changed to NUMBER OFF (from NUMBER ON STD).
==MSG>          Data does not have valid standard numbers.
==MSG> -CAUTION- Profile changed to CAPS ON (from CAPS OFF) because the
==MSG>          data does not contain any lower case characters.
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 //PREPARE JOB ( ),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID
000002 //ASSEMB EXEC DFHMAPS,MAPNAME='S01SET',RMODE=24
000003 //SYSUT1 DD *
000004 S01SET DFHMSD TYPE=MAP,MODE=INOUT,LANG=C,STORAGE=AUTO,TIOAPFX=YES
000005 *
000006 LISTE DFHMDF SIZE=(24,80),CTRL=(PRINT,FREEKB)
000007 DFHMDF POS=(9,13),ATTRB=(ASKIP,NORM),LENGTH=20, X
000008          INITIAL='VORNAME'
000009 DFHMDF POS=(9,34),ATTRB=(ASKIP,NORM),LENGTH=20, X
000010          INITIAL='NACHNAME'
000011 VNAME1 DFHMDF POS=(11,13),ATTRB=(ASKIP,NORM),LENGTH=20
000012 NNAME1 DFHMDF POS=(11,34),ATTRB=(ASKIP,NORM),LENGTH=20
Command ==>          Scroll ==> PAGE
F1=Help          F3=Exit          F5=Rfind          F6=Rchange          F12=Cancel

```

Dies ist das vollständige BMS Programm nach Fertigstellung. Es umfaßt 2 Panels Mit den F8 bzw. F7 Tasten scrollen wir zwischen den beiden Panels hin und her.

Zeilen 7 bis 10 definiert eine Überschrift, die aus 2 Feldern besteht. Die beiden Felder werden mit den Werten VORNAME und NACHNAME initialisiert.

Zeilen 11 bis 18 definieren 8 Felder, welche die Vornamen und Nachnamen von 4 Personen aufnehmen sollen, welche wir aus unserer DB2 Datenbank auslesen.

```

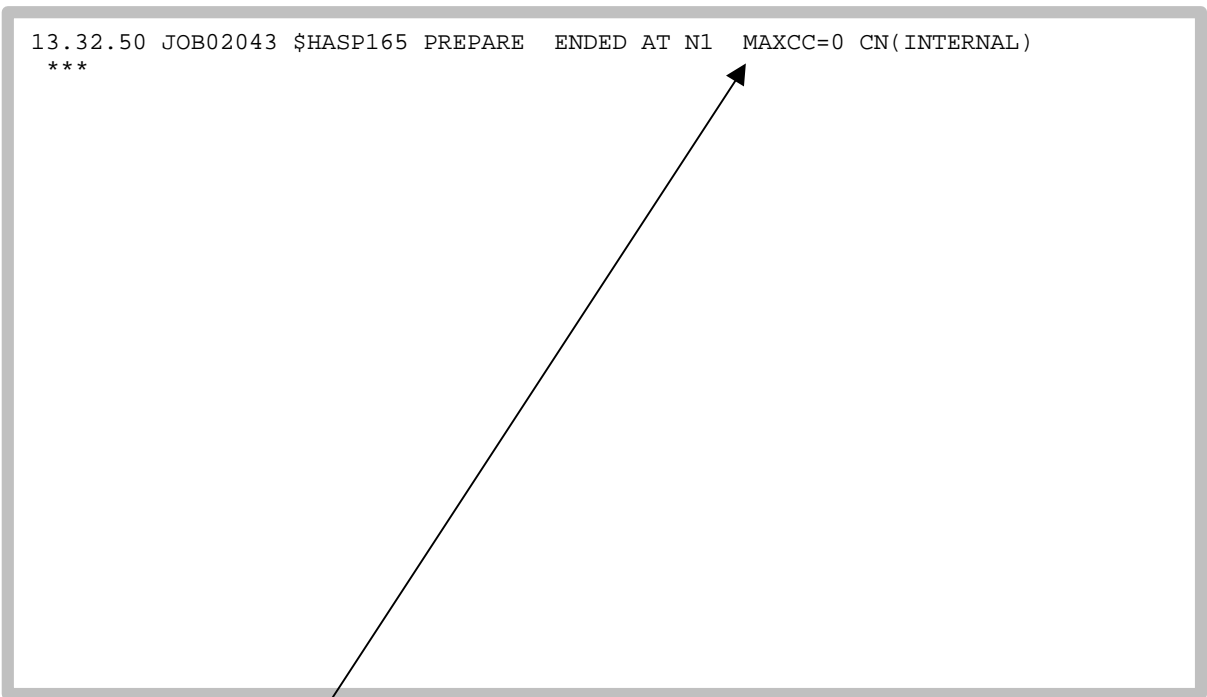
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          SPRUTH.CICSDB2.TEST01(PRAKSET) - 01.01          Columns 00001 00072
000013 VNAME2 DFHMDF POS=(12,13),ATTRB=(ASKIP,NORM),LENGTH=20
000014 NNAME2 DFHMDF POS=(12,34),ATTRB=(ASKIP,NORM),LENGTH=20
000015 VNAME3 DFHMDF POS=(13,13),ATTRB=(ASKIP,NORM),LENGTH=20
000016 NNAME3 DFHMDF POS=(13,34),ATTRB=(ASKIP,NORM),LENGTH=20
000017 VNAME4 DFHMDF POS=(14,13),ATTRB=(ASKIP,NORM),LENGTH=20
000018 NNAME4 DFHMDF POS=(14,34),ATTRB=(ASKIP,NORM),LENGTH=20
000019 DFHMSD TYPE=FINAL
000020          END
000021 /*
000022 //
***** ***** Bottom of Data *****

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

```

Wir geben "sub" auf der Kommandozeile ein. Zusätzlich zu dem übersetzten Programm wird in dem Member WGT.LIB(WGTSET1) ein Template für unser Business Logik Programm (in C) abgespeichert.

```
13.32.50 JOB02043 $HASP165 PREPARE ENDED AT N1 MAXCC=0 CN(INTERNAL)
***
```



Wir warten, bis JES unser BMS Programm übersetzt hat (30 - 60 Sekunden). Enter produziert das hier gezeigte Panel. MAXCC=0 bestätigt, daß die Übersetzung erfolgreich war.

Enter bringt uns zurück zum vorherigen screen

F3, F3

Ansehen die Members von Spruth.lib.

Wir wechseln zu dem Partitioned Data Set SPRUTH.LIB. Dort existiert jetzt das während der Übersetzung erstellte Member SPRUTH.LIB(S01SET). Wir schauen uns SPRUTH.LIB(WGTSET1) an.

```

union
{
struct {
char          dfhms1Y12";
short int     vnam1l;
char          vnam1f;
char          vnam1iY20";
short int     nnam1l;
char          nnam1f;
char          nnam1iY20";
short int     vnam2l;
char          vnam2f;
char          vnam2iY20";
short int     nnam2l;
char          nnam2f;
char          nnam2iY20";

short int     vnam3l;
char          vnam3f;
char          vnam3iY20";
short int     nnam3l;
char          nnam3f;
char          nnam3iY20";
short int     vnam4l;
char          vnam4f;
char          vnam4iY20";
short int     nnam4l;
char          nnam4f;
char          nnam4iY20";
} listei;

struct {
char          dfhms2Y12";
short int     dfhms3;
char          vnam1a;
char          vnam1oY20";

short int     dfhms4;
char          nnam1a;
char          nnam1oY20";
short int     dfhms5;
char          vnam2a;
char          vnam2oY20";
short int     dfhms6;
char          nnam2a;
char          nnam2oY20";
short int     dfhms7;
char          vnam3a;
char          vnam3oY20";
short int     dfhms8;
char          nnam3a;
char          nnam3oY20";
short int     dfhms9;
char          vnam4a;
char          vnam4oY20";
short int     dfhms10;

char          nnam4a;
char          nnam4oY20";
} listeo;

} liste;

```

Dies ist der Code von SPRUTH.LIB(WGTSET1). Er erstreckt sich über 4 Panels. Das Member enthält eine Union, die aus 2 Structures besteht. Wir verwenden es als Vorlage (Template) für die von uns als C Programm zu erstellende Business Logik

Wir rufen erneut das Edit Entry Panel auf.


```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          SPRUTH.CICSDB2.TEST01(TESTSP1) - 01.01          Columns 00001 00072
***** ***** Top of Data *****
==MSG> -CAUTION- Profile changed to CAPS OFF (from CAPS ON) because data
==MSG>          contains lower case characters.
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 #include <memory.h>
000002 #include <stdlib.h>
000003 #include </'SPRUTH.LIB(S01SET)'\>
000004
000005 EXEC SQL INCLUDE SQLCA;
000006 EXEC SQL BEGIN DECLARE SECTION;
000007 char vnameÝ20";
000008 char nnameÝ20";
000009 EXEC SQL END DECLARE SECTION;
000010
000011 main()
000012 {
000013     EXEC SQL DECLARE C1 CURSOR FOR
000014         SELECT VNAME,NNAME FROM SPRUTH.TABSP1;
Command ===>          Scroll ===> PAGE
    F1=Help          F3=Exit          F5=Rfind          F6=Rchange  F12=Cancel

```

Dies ist das vollständige C Programm nach Fertigstellung. Es umfaßt 2 Panels Mit den F8 bzw. F7 Tasten scrollen wir zwischen den beiden Panels hin und her.

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          SPRUTH.CICSDB2.TEST01(TESTSP1) - 01.01          Columns 00001 00072
000015 EXEC SQL OPEN C1;
000016 EXEC SQL FETCH C1 INTO :vname, :nname;
000017 memcpy(liste.listei.vnam1i,vname,20);
000018 memcpy(liste.listei.nnam1i,nname,20);
000019 EXEC SQL FETCH C1 INTO :vname, :nname;
000020 memcpy(liste.listei.vnam2i,vname,20);
000021 memcpy(liste.listei.nnam2i,nname,20);
000022 EXEC SQL FETCH C1 INTO :vname, :nname;
000023 memcpy(liste.listei.vnam3i,vname,20);
000024 memcpy(liste.listei.nnam3i,nname,20);
000025 EXEC SQL FETCH C1 INTO :vname, :nname;
000026 memcpy(liste.listei.vnam4i,vname,20);
000027 memcpy(liste.listei.nnam4i,nname,20);
000028 EXEC SQL CLOSE C1;
000029
000030 EXEC CICS SEND MAP("liste") MAPSET("s01set") ERASE;
000031
000032
000033 }
Command ===>          Scroll ===> PAGE
    F1=Help          F3=Exit          F5=Rfind          F6=Rchange  F12=Cancel

```

In Zeile 000007 und Zeile 000008 von WGT.CICSDB2.TEST01(TESTSP1) fällt das Sonderzeichen „Ý“ auf. Dies hat etwas mit dem „eckige Klammern“ Problem bei der Erstellung von C-Programmen zu tun.

Wir erinnern uns: TSO speichert alle Daten im EBCDIC Format. Im EBCDIC format gibt es 2 Darstellungen für die eckigen Klammern, nämlich

Character	ASCII hex	Proper EBCDIC hex	Improper EBCDIC hex
Left Square ([)	x'5B'	x'AD'	x'BA'
Right Square (])	x'5D'	x'BD'	x'BB'

Der 3270 Emulator wird häufig hex BA und hex BB als exkige Klammern " [" und "] " darstellen, während hex AD und hex BD als **⌈** und **⌋** dargestellt werden.

Wichtig ist, das Problem zu verstehen. Es gibt mehrere Möglichkeiten es anzugehen. Das Primitivverfahren ist folgendes:

Bei der Programmeingabe normal die Symbole " [" und "] " verwenden. Diese werden dann fälschlicherweise als x'BA und x'BB abgespeichert.

Nach Fertigstellung der Programmeingabe zwei globale ISPF "Change" Kommandos eingeben. Dies erfolgt durch Eingabe in der Kommandozeile:

```
C [ x'ad' all
C ] x'bd' all
```

Statt "c" kann auch "change" eingegeben werden.

Während der Übersetzung des Mapsets SPRUTH.CICSDB2.TEST01(PRAKSET) wurde ein Member S01SET im Data Set SPRUTH.LIB erstellt, der ein Template für unser C- Programm enthält. . In Übung 3 hatten wir das Template manuell in unser C - Programm kopiert. Im vorliegenden Fall gehen wir anders vor.

Zeile 000003 unseres C - Programms enthält das Statement

```
#include < //SPRUTH.LIB(S01SET)>
```

Es tritt an Stelle des manuellen Kopiervorgangs.

Nach Fertigstellung des Programms kehren wir zum Edit Entry Panel zurück.


```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          SPRUTH.CICSDB2.TEST01(PCOMPSQL) - 01.02          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 //DB2PCOMP JOB ( ),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
000002 //          TIME=1440
000003 //PCOMP      EXEC PROC=DSNZEY
000004 //DBRMLIB   DD DSN=SPRUTH.DBRMLIB.DATA(TESTSP1),DISP=OLD
000005 //SYSCIN     DD DSN=SPRUTH.CICSDB2.TEST01(OUT),DISP=SHR
000006 //SYSLIB     DD DSN=SPRUTH.CICSDB2.TEST01,DISP=SHR
000007 //SYSIN     DD DISP=SHR,DSN=SPRUTH.CICSDB2.TEST01(TESTSP1)
***** ***** Bottom of Data *****

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

```

Wir erstellen das hier dargestellte JCL Script zum Aufruf des EXEC SQL Precompilers.

SPRUTH.DBRMLIB.DATA enthält bis hierher keine Members. Nach der Ausführung des JCL Scriptes hat der Precompiler einen Member SPRUTH.DBRMLIB.DATA(TESTSP1) angelegt.

Auf der Kommandozeile sub (für submit) eingeben. Wir warten die Ausführung des JES Jobs ab. Enter

```

17.12.04 JOB02051 $HASP165 DB2PCOMP ENDED AT N1  MAXCC=0 CN(INTERNAL)
***

```

Erfolgreich ausgeführt.

Enter


```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          SPRUTH.CICSDB2.TEST01(STARTSQL) - 01.04          Columns 00001 00072
000017 //DBRMLIB DD DISP=OLD,DSN=SPRUTH.DBRMLIB.DATA(TESTSP1)
000018 //SYSPRINT DD SYSOUT=*
000019 //SYSPRT DD SYSOUT=*
000020 //SYSUDUMP DD SYSOUT=*
000021 //SYSTSIN DD *
000022 DSN S(DBA1)
000023 BIND PLAN(SPR) MEMBER(TESTSP1) ACTION(REP) RETAIN ISOLATION(CS)
000024 END
000025 //*****
000026 /* GRANT
000027 //*****
000028 //GRANT EXEC PGM=IKJEFT01
000029 //STEPLIB DD DISP=SHR,DSN=DSN510.SDSNLOAD
000030 //SYSPRINT DD SYSOUT=*
000031 //SYSPRT DD SYSOUT=*
000032 //SYSUDUMP DD SYSOUT=*
000033 //SYSTSIN DD *
000034 DSN SYSTEM(DBA1)
000035 RUN PROGRAM(DSNTIAD) PLAN(DSNTIA51) -
Command ==>
F1=Help      F3=Exit      F5=Rfind      F6=Rchange   F12=Cancel   Scroll ==> PAGE

```

Panel # 2. Der SQL Precompiler Lauf hat im Data Set SPRUTH.DBRMLIB.DATA ein Member TESTSP1 angelegt.

Die Ausführung unseres Programms TESTSP1 unter CICS benötigt einen Zeiger auf die anzusprechende Datenbank Tabelle (als PLAN bezeichnet). Wir geben diesem Zeiger den Namen SPR.

F8 Taste

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          SPRUTH.CICSDB2.TEST01(STARTSQL) - 01.04          Columns 00001 00072
000036 LIBRARY('DSN510.RUNLIB.LOAD')
000037 END
000038 //SYSIN DD *
000039 GRANT EXECUTE ON PLAN SPR TO PUBLIC
000040 /*
***** ***** Bottom of Data *****
Command ==>
F1=Help      F3=Exit      F5=Rfind      F6=Rchange   F12=Cancel   Scroll ==> PAGE

```

Panel # 3. Die Referenz auf Tabelle (Plan) SPR taucht nochmals auf.

Sie können für ähnliche Aufgaben dieses JCL Script immer wieder verwenden, indem Sie lediglich an den durch Pfeile gekennzeichneten Stellen ihre eigenen Programmnamen eingeben.

```

//CICSPRE JOB ( ),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//          TIME=1440
//*****
//* TRANSL/COMP/LINKEDIT
//*****
//COMP      EXEC PROC=CTOCICS,REG=0M
//TRN.SYSIN DD DISP=SHR,DSN=SPRUTH.CICSDB2.TEST01(OUT)
//LKED.SYSIN DD *
//          INCLUDE DB2LOAD(DSNCLI)
//          NAME TESTSP1(R)
//*****
//* BIND
//*****
//BIND      EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=DSN510.SDSNEXIT
//          DD DISP=SHR,DSN=DSN510.SDSNLOAD

//DBRMLIB DD DISP=OLD,DSN=SPRUTH.DBRMLIB.DATA(TESTSP1)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
//          DSN S(DBA1)
//          BIND PLAN(SPR) MEMBER(TESTSP1) ACTION(REP) RETAIN ISOLATION(CS)
END
//*****
//* GRANT
//*****
//GRANT     EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=DSN510.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
//          DSN SYSTEM(DBA1)
//          RUN PROGRAM(DSNTIAD) PLAN(DSNTIA51) -

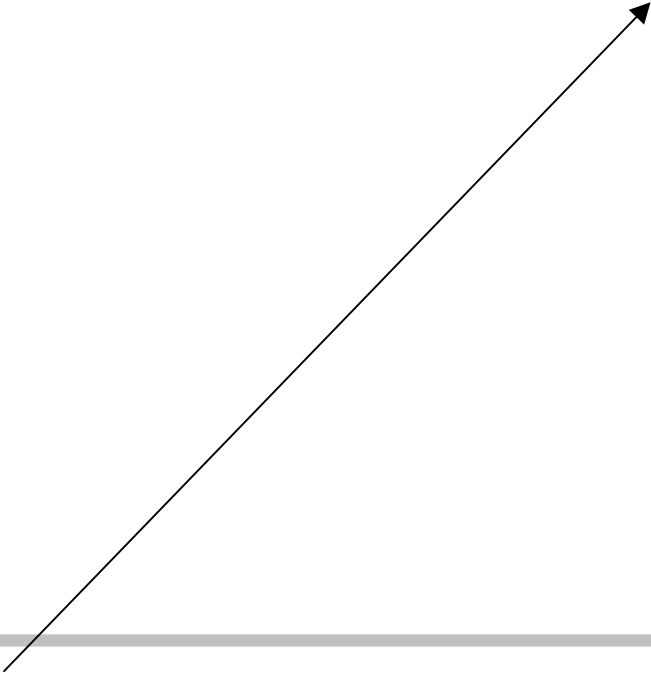
LIBRARY('DSN510.RUNLIB.LOAD')
//          END
//SYSPRINT DD *
//          GRANT EXECUTE ON PLAN SPR TO PUBLIC
//          /*

```

Wir geben "sub" auf der Kommandozeile ein und warten, bis JES den Job ausgegeben hat.

Enter

```
18.53.21 JOB02053 $HASP165 CICS PRE ENDED AT N1 MAXCC=4 CN(INTERNAL)
***
```



MAXCC=4 bedeutet, daß der Compile und Link Lauf erfolgreich durchgeführt wurde.

Wir haben nun alle Programme für unsere CICS - DB2 Transaktion erstellt. Als nächsten Schritt müssen sie in dem CICS Subsystem installiert werden. Hierzu öffnen wir eine weitere OS/390 Session.

```
TCPIP MSG10 ==> SOURCE DATA SET = SYS1.LOCAL.VTAMLST(USSTCPIP)

03/01/01                W E L C O M E T O                19:20:39

      SSSSSS  //  3333333  9999999  0000000
SS      //  33  33  99  99  00  00
SS      //  33  99  99  00  00
SSSS   //  33333  9999999  00  00
      SS //  33      99  00  00
      SS //  33  33  99  99  00  00
SSSSSS //  3333333  9999999  0000000

YOUR TERMINAL NAME IS :                YOUR IP ADDRESS IS : 217.002.103.061

      APPLICATION DEVELOPMENT SYSTEM
      OS/390 RELEASE 2.7.0

===> ENTER "L " FOLLOWED BY THE APPLID YOU WISH TO LOGON TO.  EXAMPLE "L TSO"
      FOR TSO/E OR "L C001" FOR THE CICS001 CICS APPLICATION.

L C001
```

Wir loggen uns mit L C001 ein. Enter

```
CICS C001 A06C001      IBM DEMONSTRATION SYSTEM 24:00:00
```

```
*****\  *****\  *****\  *****\  
*****\  *****\  *****\  *****\  
**\\\\\\**\  **\\\\\\  **\\\\\\**\  **\\\\\\**\  
**\      \\\  **\      \\\  **\      \\\  **\      \\  
**\      \\\  **\      \\\  **\      \\\  *****\  
**\      \\\  **\      \\\  **\      \\\  *****\  
**\      \\\  **\      \\\  **\      \\\  \\\\\\\**\  
**\      \\\  **\      \\\  **\      \\\  **\      \\  
*****\  *****\  *****\  *****\  
*****\  *****\  *****\  *****\  
\\\\\\\\\\  \\\\\\\\\  \\\\\\\\\  \\\\\\\\\  TM
```

```
DFHAC2001 03/01/01 21:25:38 A06C001 Transaction ' ' is not recognized. Check  
that the transaction name is correct. CEDA DISPLAY GROUP(*)
```

Enter. Das CICS Entry Panel erscheint. Nochmals enter. Eine Fehlermeldung erscheint. Wir geben den CEDA DISPLAY GROUP(*) Befehl ein.

Enter

```
CEDA DEFINE MAPSET(S01SET) GROUP(SPRUTH1)  
ENTER COMMANDS  
GROUP  
AOR2TOR  
ARTT  
ATC  
CBPS  
CEE  
CICREXX  
CSQ  
CSQCKB  
CSQSAMP  
CTA1TCP  
C001EZA  
C001TCP  
DAVIN4  
DAVIN8  
DAVIN85  
DAVIN9  
+ DAVIN94  
  
RESULTS: 1 TO 17  
PF 1 HELP      3 END 4 TOP 5 BOT 6 CRSR 7 SBN 8 SFH 9 MSG 10 SB 11 SF 12 CNCL  
SYSID=C001 APPLID=A06C001  
TIME: 00.00.00 DATE: 01.060
```

Der CEDA DISPLAY GROUP(*) Befehl zeigt alle bisher vorhandenen Gruppen an.

Wir definieren zunächst unser BMS Programm mit dem Namen S01SET1 für die neue Group SPRUTH1.

Dieser und die nächsten Schritte sind praktisch eine Wiederholung von Übung Nr. 3

Enter

```

CEDA DEFINE MAPSET(S01SET) GROUP(SPRUTH1)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE Mapset( S01SET  )
Mapset      : S01SET
Group       : SPRUTH1
Description ==>
REsident    ==> No                No | Yes
USAge       ==> Normal            Normal | Transient
USElpacopy  ==> No                No | Yes
Status      ==> Enabled           Enabled | Disabled
RS1         : 00                  0-24 | Public

I New group SPRUTH1 created.

DEFINITION SUCCESSFUL                            SYSID=C001 APPLID=A06C001
TIME: 00.00.00 DATE: 01.060
PF 1 HELP 2 COM 3 END                            6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Die Installation war erfolgreich, und die neue Gruppe wurde erstellt.

```

CEDA DEFINE PROG(TESTSP1) GROUP(SPRUTH1)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE Mapset( S01SET  )
Mapset      : S01SET
Group       : SPRUTH1
Description ==>
REsident    ==> No                No | Yes
USAge       ==> Normal            Normal | Transient
USElpacopy  ==> No                No | Yes
Status      ==> Enabled           Enabled | Disabled
RS1         : 00                  0-24 | Public

I New group SPRUTH1 created.

DEFINITION SUCCESSFUL                            SYSID=C001 APPLID=A06C001
TIME: 00.00.00 DATE: 01.060
PF 1 HELP 2 COM 3 END                            6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Als nächstes wird unser C Programm definiert

Enter

```

CEDA DEFINE PROG(TESTSP1) GROUP( SPRUTH1)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFine PROGram( TESTSP1 )
  PROGRAM      : TESTSP1
  Group       : SPRUTH1
  Description  ==>
  Language    ==> Le370          CObol | Assembler | Le370 | C | Pli
  REload     ==> No             No | Yes
  RESident   ==> No             No | Yes
  USAge      ==> Normal         Normal | Transient
  USElpacopy ==> No             No | Yes
  Status     ==> Enabled        Enabled | Disabled
  RSl        : 00               0-24 | Public
  CEdf       ==> Yes            Yes | No
  DAtalocation ==> Below        Below | Any
  EXECKey    ==> User           User | Cics
  COncurrency ==> Quasirent     Quasirent | Threadsafe
  REMOTE ATTRIBUTES
  Dynamic    ==> No             No | Yes
+ REMOTESystem ==>

                                     SYSID=C001 APPLID=A06C001
  DEFINE SUCCESSFUL                    TIME: 00.00.00 DATE: 01.060
PF 1 HELP 2 COM 3 END                  6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Le370 als Language (in Wirklichkeit als Laufzeitumgebung) eingeben.

Enter

Die Nachricht DEFINE SUCCESSFUL erscheint

F3 Taste

```

CEDA DEFINE TRANS(SP01) GROUP( SPRUTH1)
STATUS: SESSION ENDED

```

Als letztes müssen wir die Bezeichnung der neuen Transaktion definieren. Wir wählen auch hierfür den Namen SP01. Es könnte natürlich auch ein beliebiger anderer Name sein, solange er aus 4 Zeichen besteht. Enter

```

DEFINE TRANS(SP01) GROUP( SPRUTH1)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE TRANSAction( SP01 )
  TRANSAction  ==> SP01
  Group        ==> SPRUTH1
  Description   ==>
  PROGRAM      ==> TESTSP1
  TWasize      ==> 00000          0-32767
  PROFILE      ==> DFHCICST
  PARTITIONSET ==>
  STATUS       ==> Enabled      Enabled | Disabled
  PRIMEDSIZE   : 00000          0-65520
  TASKDATALOC ==> Below        Below | Any
  TASKDATAKEY  ==> User        User | Cics
  STORAGECLEAR ==> No          No | Yes
  RUNAWAY      ==> System      System | 0 | 500-2700000
  SHUTDOWN     ==> Disabled    Disabled | Enabled
  ISOLATE      ==> Yes         Yes | No
  BREXIT       ==>
+ REMOTE ATTRIBUTES
  S PROGRAM OR REMOTESYSTEM MUST BE SPECIFIED.
                                           SYSID=C001 APPLID=A06C001

PF 1 HELP 2 COM 3 END                    6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

In die Zeile PROGRAM Zeile TESTSP1 eingeben

Enter

```

OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE TRANSAction( SP01 )
  TRANSAction  : SP01
  Group        : SPRUTH1
  Description   ==>
  PROGRAM      ==> TESTSP1
  TWasize      ==> 00000          0-32767
  PROFILE      ==> DFHCICST
  PARTITIONSET ==>
  STATUS       ==> Enabled      Enabled | Disabled
  PRIMEDSIZE   : 00000          0-65520
  TASKDATALOC ==> Below        Below | Any
  TASKDATAKEY  ==> User        User | Cics
  STORAGECLEAR ==> No          No | Yes
  RUNAWAY      ==> System      System | 0 | 500-2700000
  SHUTDOWN     ==> Disabled    Disabled | Enabled
  ISOLATE      ==> Yes         Yes | No
  BREXIT       ==>
+ REMOTE ATTRIBUTES
                                           SYSID=C001 APPLID=A06C001
  DEFINE SUCCESSFUL                          TIME: 00.00.00 DATE: 01.060
PF 1 HELP 2 COM 3 END                    6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

War erfolgreich

F3 Taste

```
CEDA INSTALL GROUP( SPRUTH1 )
STATUS:  SESSION ENDED
```

Nachdem unsere BMS MAP, unser C Programm und unsere Transaktionsbezeichnung definiert worden sind, wird nun alles in unserer Gruppe SPRUTH1 installiert

Enter

```
INSTALL GROUP( SPRUTH1 )
OVERTYPE TO MODIFY
  CEDA Install
    All
    Connection ==>
    DB2Conn    ==>
    DB2Entry   ==>
    DB2Tran    ==>
    DOctemplate ==>
    Enqmodel   ==>
    File       ==>
    Journalmodel ==>
    LSrpool    ==>
    Mapset     ==>
    PARTitionset ==>
    PARTner    ==>
    PROCesstype ==>
    PROFile    ==>
    PROGram    ==>
+ Requestmodel ==>

                                SYSID=C001 APPLID=A06C001
                                TIME: 00.00.00 DATE: 01.060
INSTALL SUCCESSFUL
PF 1 HELP 3 END                6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

Hat geklappt

F3 Taste

```
SP01  
STATUS:  SESSION ENDED
```

In Übung 3 waren wir damit mit der Definition und Installation unserer Transaktion fertig. Also, versuchen wir es doch einmal. Wir rufen unsere Transaktion mit der Bezeichnung SP01 auf, indem wir den Namen in die CICS Kommandozeile eingeben

Enter

```
DFHAC2220 20:49:29 A06C001 The coordinator system has indicated that the  
current unit of work is to be backed out. Transaction SP01 has been  
abnormally terminated with abend ASP3.
```

Wir erhalten eine häßliche Fehlermeldung

Die Beschreibung zur Fehlermeldung ASP3 findet sich in einem Online Handbuch:

<http://www.s390.ibm.com/bookmgr-cgi/bookmgr.cmd/BOOKS/DFHWG400/2%2e3%2e606?ACTION=MATCHES&REQUEST=asp3&TYPE=FUZZY&SHELF=&searchTopic=TOPIC&searchText=TEXT&searchIndex=INDEX&rank=RANK&ScrollTOP=FIRSTHIT#FIRSTHIT>

Wir suchen nach dem Fehlercode ASP3 und finden den folgenden Eintrag:

Explanation: The abnormal termination occurs because a remote system on which the unit of work depends fails to take a syncpoint. The transaction cannot commit its changes until all coupled systems to which function has been transmitted also commit. This may be because the syncpoint protocol for transaction to transaction has been violated by failing to be in send mode for all sessions for which syncpoint has not been received.

User Response:

Check why the remote system failed to respond to the request.

Leicht irreführend. Wir erinnern uns: TSO, CICS und DB2 sind alle separate OS/390 Subsysteme, die in getrennten virtuellen Adressenräumen laufen. Die CICS Group SPRUTH1 benötigt eine Definition unserer Datenbank und Datenbanktabelle.

```
DFHAC2220 20:56:06 A06C001 The coordinator system has indicated that the
DFHAC2001 03/02/01 20:56:29 A06C001 Transaction ' ' is not recognized. Check
that the transaction name is correct.  CEDA DEFINE DB2ENTRY
```

Die Definition erfolgt mit dem CEDA Kommando DEFINE DB2ENTRY

Enter

```

DEFINE DB2ENTRY
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA Define DB2Entry(                               )
  DB2Entry    ==>
  Group       ==>
  Description  ==>
THREAD SELECTION ATTRIBUTES
  Transid     ==>
THREAD OPERATION ATTRIBUTES
  ACountrec   ==> None           None | TXid | TAsk | Uow
  AUTHID      ==>
  AUTHType    ==>               Userid | Opid | Group | Sign | Term
                                   | TX
  DRollback   ==> Yes           Yes | No
  PLAN        ==>
  PLANExitname ==>
  PRiority    ==> High         High | Equal | Low
  PROtectnum  ==> 0000         0-2000
  THREADLimit ==>              0-2000
  THREADWait  ==> Pool         Pool | Yes | No
MESSAGES: 2 SEVERE

                                   SYSID=C001 APPLID=A06C001

PF 1 HELP 2 COM 3 END                6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Das DEFINE DB2ENTRY Panel erscheint. Wir müssen die fehlenden Angaben eintragen

```

define DB2ENTRY
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA Define DB2Entry(                               )
  DB2Entry    ==> SP01
  Group       ==> SPRUTH1
  Description  ==>
THREAD SELECTION ATTRIBUTES
  Transid     ==> SP01
THREAD OPERATION ATTRIBUTES
  ACountrec   ==> None           None | TXid | TAsk | Uow
  AUTHID      ==>
  AUTHType    ==>               Userid | Opid | Group | Sign | Term
                                   | TX
  DRollback   ==> Yes           Yes | No
  PLAN        ==> SPR
  PLANExitname ==>
  PRiority    ==> High         High | Equal | Low
  PROtectnum  ==> 0000         0-2000
  THREADLimit ==>              0-2000
  THREADWait  ==> Pool         Pool | Yes | No
MESSAGES: 2 SEVERE

                                   SYSID=C001 APPLID=A06C001

PF 1 HELP 2 COM 3 END                6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Wir bezeichnen den DB2 Zugriff (DB2Entry) mit dem Namen SP01. Das Ganze wird Teil der Gruppe SPRUTH1. Unsere Transactions ID (TRID) ist SP01.

Erinnern wir uns: Wir hatten ein JCL Script STARTSQL erstellt, welches unser C Programm übersetzte. In diesem Script definierten wir an zwei Stellen einen Zeiger auf unsere Datenbanktabelle (Plan) mit dem Namen SPR. Hier wird jetzt für CICS die Verknüpfung zu der Datenbanktabelle hergestellt.

```

define DB2ENTRY
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE DB2ENTRY(                               )
  DB2ENTRY    ==> SP01
  Group       ==> SPRUTH1
  Description  ==>
THREAD SELECTION ATTRIBUTES
  TRANSID     ==> SP01
THREAD OPERATION ATTRIBUTES
  ACcountrec  ==> TXid ←
  AUTHID      ==>
  AUTHType    ==> Sign ←
                                     Userid | Opid | Group | Sign | TTerm
                                     | TX
  DRollback   ==> Yes                 Yes | No
  PLAN        ==> SPR
  PLANExitname ==>
  PRiority    ==> High                 High | Equal | Low
  PROtectnum  ==> 0000                 0-2000
  THREADLimit ==> 0003 ←               0-2000
  THREADWait  ==> Yes ←               Pool | Yes | No
  MESSAGES: 2 SEVERE

                                     SYSID=C001 APPLID=A06C001

PF 1 HELP 2 COM 3 END                        6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Wir geben weiterhin die gekennzeichneten 4 Parameter ein.

Enter

```

OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE DB2ENTRY( SP01                          )
  DB2ENTRY    : SP01
  Group       : SPRUTH1
  Description  ==>
THREAD SELECTION ATTRIBUTES
  TRANSID     ==> SP01
THREAD OPERATION ATTRIBUTES
  ACcountrec  ==> TXid                 None | TXid | TAsk | Uow
  AUTHID      ==>
  AUTHType    ==> Userid               Userid | Opid | Group | Sign | TTerm
                                     | TX
  DRollback   ==> Yes                 Yes | No
  PLAN        ==> SPR
  PLANExitname ==>
  PRiority    ==> High                 High | Equal | Low
  PROtectnum  ==> 0000                 0-2000
  THREADLimit ==> 0003                 0-2000
  THREADWait  ==> Yes                 Pool | Yes | No

                                     SYSID=C001 APPLID=A06C001
                                     TIME: 00.00.00 DATE: 01.061
DEFINE SUCCESSFUL
PF 1 HELP 2 COM 3 END                        6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Die Definition war erfolgreich

F3 Taste

```
CEDA INSTALL GROUP( SPRUTH1 )
STATUS:  SESSION ENDED
```

Der gleiche Prozess wie vorher: Die Änderung muß installiert werden

Enter

```
INSTALL GROUP( SPRUTH1 )
OVERTYPE TO MODIFY
CEDA Install
All
Connection ==>
DB2Conn ==>
DB2Entry ==>
DB2Tran ==>
DOctemplate ==>
Enqmodel ==>
File ==>
Journalmodel ==>
LSrpool ==>
Mapset ==>
PARTitionset ==>
PARTner ==>
PROcesstype ==>
PROfile ==>
PROgram ==>
+ Requestmodel ==>

                                SYSID=C001 APPLID=A06C001
INSTALL SUCCESSFUL              TIME: 00.00.00 DATE: 01.061
PF 1 HELP                        3 END          6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

OK

F3 Taste

```
SP01
STATUS:  SESSION ENDED
```

So, jetzt geben wir unsere Transaktion mit dem Namen SP01 nochmals ein -

Enter

```
VORNAME          NACHNAME
HEINO            BAUER
BORIS            FAERBER
SEBASTIAN        RICHTER
FRITZ            SCHULTE
```

- und werden mit dem erwarteten Ergebnis belohnt. Herzlichen Glückwunsch.

Puh, das war ein schönes Stück Arbeit, bei dem sich OS/390 von einer seiner unangenehmsten Seiten zeigte. Wenn Sie bis hierhin vorgedrungen sind, haben Sie sich bereits ein beachtliches Verständnis über die Arbeitsweise von OS/390 angeeignet. Sie haben gelernt, wie man eine Transaktion programmiert, wie man eine relationale DB2 Datenbank anlegt und wie man aus dem Transaktionscode auf die Datenbank zugreift.

Im nächsten Abschnitt schauen wir uns an, wie man eine OS/390 Transaktion internet fähig macht. Bis dahin.

Falls Sie in Probleme hineinlaufen: Hier gibt es die Online Handbuecher:

<http://www.s390.ibm.com/bookmgr-cgi/bookmgr.cmd/library>

Der CICS DB Guide befindet sich unter:

<http://www.s390.ibm.com/bookmgr-cgi/bookmgr.cmd/BOOKS/DFHCATK4/CCONTENTS>

Die Beschreibung zu Fehlermeldungen:

<http://www.s390.ibm.com/bookmgr-cgi/bookmgr.cmd/BOOKS/DFHWG400/2%2e3%2e606?ACTION=MATCHES&REQUEST=asp3&TYPE=FUZZY&SHELF=&searchTopic=TOPIC&searchText=TEXT&searchIndex=INDEX&rank=RANK&ScrollTOP=FIRSTHIT#FIRSTHIT>

oder im Buch CICS Messages and Codes:

<http://www.s390.ibm.com/bookmgr-cgi/bookmgr.cmd/BOOKS/DFHWG400/CCONTENTS>