

Tutorial 3

CICS (Assembler-Version)

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

Ziel dieser Aufgabe ist es, in CICS eine eigene Transaktion zu installieren und diese anschließend auszuführen. Der Business Logic-Teil soll in Assembler geschrieben werden.

Für eine weiterführende Beschäftigung mit CICS empfiehlt sich das Buch "Einführung in z/OS und OS/390" (P. Herrmann et. al.), das in der zweiten, korrigierten Auflage im Jahr 2004 im Oldenbourg Verlag (München, Wien) erschienen ist. Es enthält ein Kapitel, das die Eigenschaften des CICS-Tranaktions-Monitors beschreibt.

Aufgabe: Arbeiten Sie nachfolgendes Tutorial durch. Beachten Sie, dass "Prakt20" in allen Datasetnamen durch Ihre User-ID auf dem Leipziger S/390-Mainframe ersetzt werden muß.

TSO ist ein z/OS-Subsystem, CICS ein weiteres z/OS-Subsystem. Jedes der beiden Subsysteme hat eine eigene Benutzerschnittstelle (eine eigene Shell). Um eine CICS-Anwendung zu erstellen, müssen wir mit beiden Subsystemen arbeiten: Mit TSO, um die Anwendung zu erzeugen, und mit CICS, um die Anwendung (unter dem CICS-Subsystem) auszuführen. Da z/OS ein Multi-User-Betriebssystem ist (multisession-fähig), können wir gleichzeitig eine TSO-Session und eine CICS-Session auf unserem Arbeitsplatzrechner laufen lassen. Jede Session läuft in einem eigenen Fenster.

Wir starten unseren 3270-Emulator zunächst für eine TSO-Session und loggen uns ein.

Wir öffnen den "Data Set Utility"-Screen und erzeugen (Allocate) einen neuen Partitioned Dataset: "PRAKT20.CICS.ASSEM". Dabei verwenden wir die in der nachstehenden Aufgabe angegebenen Parameter.

Außerdem brauchen wir noch einen Partitioned Dataset mit dem vorgegebenen Namen "PRAKT20.LIB", dessen Members von der Entwicklungsumgebung während der CICS-Programmentwicklung mit Daten gefüllt werden. Verwenden Sie bei dessen Anlage ebenfalls die unten angegebenen Parameter.

Dieser Name besteht aus einem "Projekt"-Begriff und einem "Group"-Begriff. Es fehlt der "Type"-Begriff. Wenn wir das Typ-Feld leer lassen, wird TSO dies nicht akzeptieren. Deshalb tragen wir den Namen 'PRAKT20.LIB' (mit Hochkommas!) in die Zeile "Data Set Name" ein. Damit wird auch dieser Dataset angelegt.

***Aufgabe:** Legen Sie, wenn noch nicht vorhanden, die Datasets "PRAKT20.CICS.ASSEM" und "PRAKT20.LIB" an. Verwenden Sie dazu folgende Parameter:*

<i>Space units</i>	<i>KILOBYTE</i>	<i>Record format</i>	<i>FB</i>
<i>Primary quantity . .</i>	<i>16</i>	<i>Record length</i>	<i>80</i>
<i>Secondary quantity</i>	<i>1</i>	<i>Block size</i>	<i>320</i>
<i>Directory blocks . .</i>	<i>2</i>	<i>Data set name type :</i>	<i>PDS</i>

Sollte ein Dataset schon vorhanden sein, empfiehlt sich ein Löschen aller Member dieses Datasets sowie ein anschließender Compress.

Unsere Anwendung besteht aus zwei Programmteilen und einem JCL-Script für die Übersetzung. Wir erstellen diese als Members in dem neuen Partitioned Dataset "PRAKT20.CICS.ASSEM".

Ein sauber strukturiertes CICS-Programm besteht aus zwei Teilen: Business Logic und Presentation Logic. Business Logic ist der Teil, in dem Berechnungen erfolgen und Daten aus/in eine(r) Datenbank gelesen/geschrieben werden. Presentation Logic ist der Teil, in dem die Ergebnisse so aufgearbeitet werden, dass sie dem Benutzer in einer ansprechenden Art auf dem Bildschirm dargestellt werden.

Business Logic wird in Sprachen wie C++, COBOL, Assembler und PL/1 geschrieben. Für die Presentation Logic gibt es viele Alternativen. Die modernste Alternative benutzt Java Server Pages und einen Web Application Server, um den Bildschirminhalt innerhalb eines Web Browsers darzustellen. Die älteste (und einfachste) Alternative verwendet das CICS BMS (Basic Mapping Support)-Subsystem. BMS-Programme werden in der BMS-Sprache geschrieben. In unserem Beispiel wird die Business Logic in Assembler und die Presentation Logic in BMS geschrieben.

Wir fangen mit dem letzteren an, rufen den "Edit Entry Panel"-Bildschirm auf und editieren ein Member "ASSMAP3" für den neu angelegten Partitioned Dataset "PRAKT20.CICS.ASSEM" (s. Abbildung 1).

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          PRAKT20.CICS.ASSEM(ASSMAP3) - 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.
000100 //PRAK020B JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
000200 //          REGION=4M
000300 //ASSEM EXEC DFHMAPS,MAPNAME='ASET020',RMODE=24
000400 //COPY.SYSUT1 DD *
000500 ASET020  DFHMSD TYPE=MAP,MODE=INOUT,LANG=ASM,STORAGE=AUTO,      X
000600          TIOAPFX=YES
000700 *          MENU MAP
000800 MAP020  DFHMDI SIZE=(24,80),CTRL=(PRINT,FREEKB)
000900          DFHMDF POS=(9,23),ATTRB=(ASKIP,NORM),LENGTH=34,      X
001000          INITIAL='WELCOME TO THE MAGIC WORLD OF CICS'
001100          DFHMDF POS=(12,33),ATTRB=(ASKIP,NORM),LENGTH=15,      X
001200          INITIAL='MEMBER PRAKT20 !'
001300          DFHMSD TYPE=FINAL
001400          END
001500 /*
001600 //
Command ==> SUB          Scroll ==> PAGE
F1=Help          F3=Exit          F5=Rfind          F6=Rchange          F12=Cancel

```

Abbildung 1: Das BMS-Programm

Das BMS-Programm enthält neben den drei BMS-Befehlen DFHMSD, DFHMDI und DFHMDF ein Mapset "ASET020", das nur eine Map "MAP020" besitzt (Userid: PRAKT20). Das BMS-Programm spezifiziert ein JCL-Script mit BMS-Befehlen für die Presentation Logic. Wir geben auf der Kommandozeile den ISPF-Befehl "SUB" ein. Es wird die Prozedur "DFHMAPS" ausgeführt.

Aufgabe: Schreiben Sie ein BMS-Programm, das einen Mapset, der unter CICS eine Ausgabe ähnlich der in Abbildung 2 dargestellten liefert, erzeugt. Die Ausgabe muß unbedingt den **Transaktionsnamen**, **"TUTORIAL 3 IN ASSEMBLER"**, das **Bearbeitungsdatum** sowie **den Namen oder die Namen der Autoren** enthalten. Ersetzen Sie in den Bezeichnern für Ihren Mapset und Ihre Map die Ziffern "020" durch die Nummer Ihres PRAKT<xx> oder PRAK<xxx>-Accounts. Übersetzen Sie anschließend das BMS-Programm. (Hinweis: Eine fehlerlose Übersetzung des BMS-Programms wird durch die Statusmeldung "MAXCC=0" beendet).

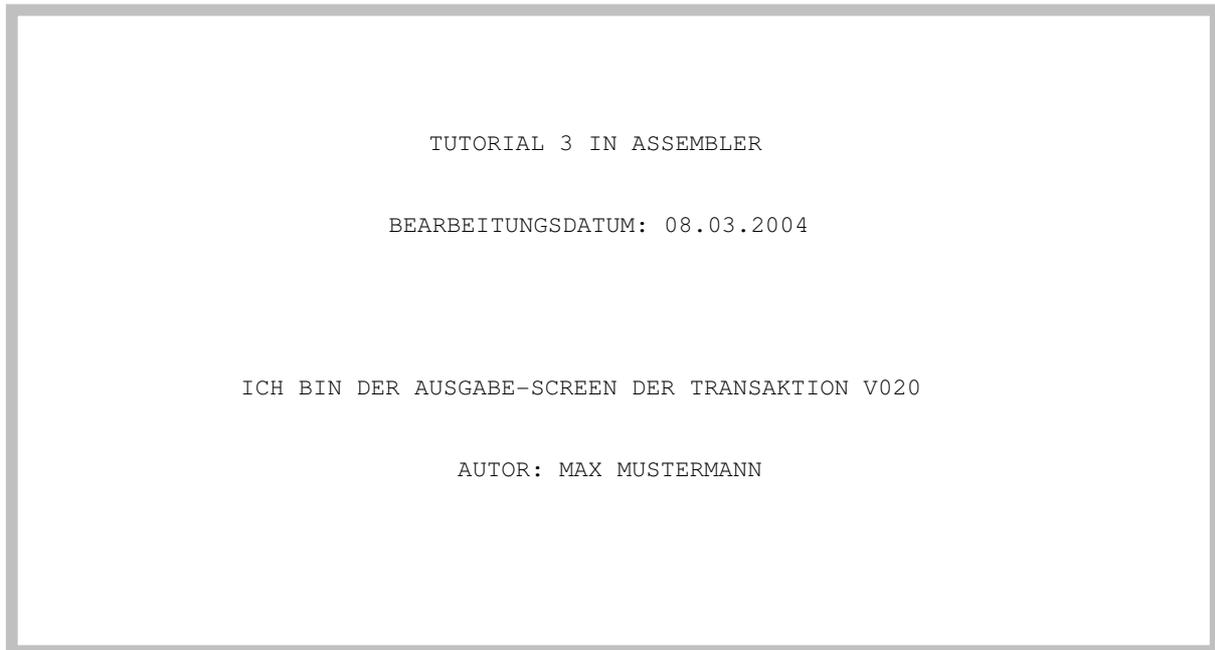


Abbildung 2: Ausgabe der Transaktion V020

Im nächsten Schritt editieren wir das Assembler-Anwendungsprogramm als Member "ASSPR020" im PDS "PRAKT20.CICS.ASSEM" (s. Abbildung 15).

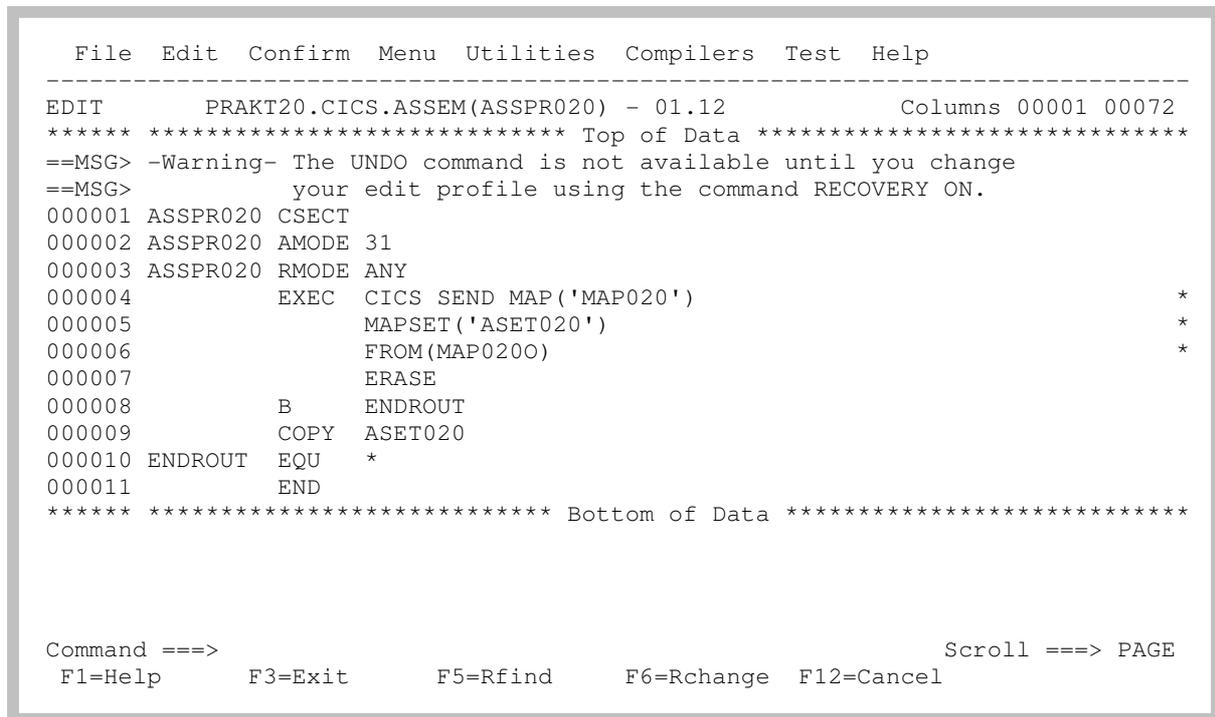


Abbildung 15: Assembler-Programm " ASSPR020"

Das Anwendungsprogramm muss nun übersetzt werden. Zu diesem Zweck wird ein weiteres JCL-File mit dem Namen "ASSSTA03" als Member im PDS "PRAKT20.CICS.ASSEM" erstellt (s. Abbildung 16). In dem Member "ASSSTA03" wird die Prozedur ASSCICS ausgeführt, die alle CICS-Befehle in Assembler-Befehle übersetzt. Anschließend wird ein ausführbares Maschinenprogramm generiert.

Auf der Kommando-Zeile des Edit-Screens geben wir ein "SUB" ein. Dadurch wird der Job "PRAKT20.CICS.ASSEM(ASSTA03) ausgeführt. Falls der Completion Code MAXCC=0 erscheint (ersichtlich mit der Eingabetaste oder F3), ist der Job erfolgreich gelaufen.

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          PRAKT20.CICS.ASSEM(ASSTA03) - 01.07          Columns 00001 00072
***** ***** Top of Data *****
==MSG> -CAUTION- Profile changed to NUMBER ON STD (from NUMBER OFF).
==MSG>          Data has valid standard numbers.
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000100 //PRAK020D JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
000200 //          REGION=4M
000400 //STEP1 EXEC ASSCICS
000500 //TRN.SYSIN DD DISP=SHR,DSN=&SYSUID..CICS.ASSEM(ASSPR020)
000600 //ASM.SYSLIB DD
000601 //          DD
000602 //          DD
000603 //          DD
000610 //          DD DSN=&SYSUID..LIB,DISP=SHR
000700 //LKED.SYSIN DD *
000710 ENTRY ASSPR020
000800 NAME ASSPR020(R)
000900 /*
***** ***** Bottom of Data *****
Command ==> SUB          Scroll ==> PAGE
F1=Help      F3=Exit      F5=Rfind      F6=Rchange  F12=Cancel

```

Abbildung 16: JCL-File "ASSTA03"

Aufgabe: Schreiben Sie das Assembler-Programm und übersetzen Sie dieses. Ersetzen Sie überall im Assembler-Programm, im JCL-File sowie im Membernamen des Assembler-Programms "ASSPR020" durch "ASSPR<Ihre S/390-UnserID-Nr.>".

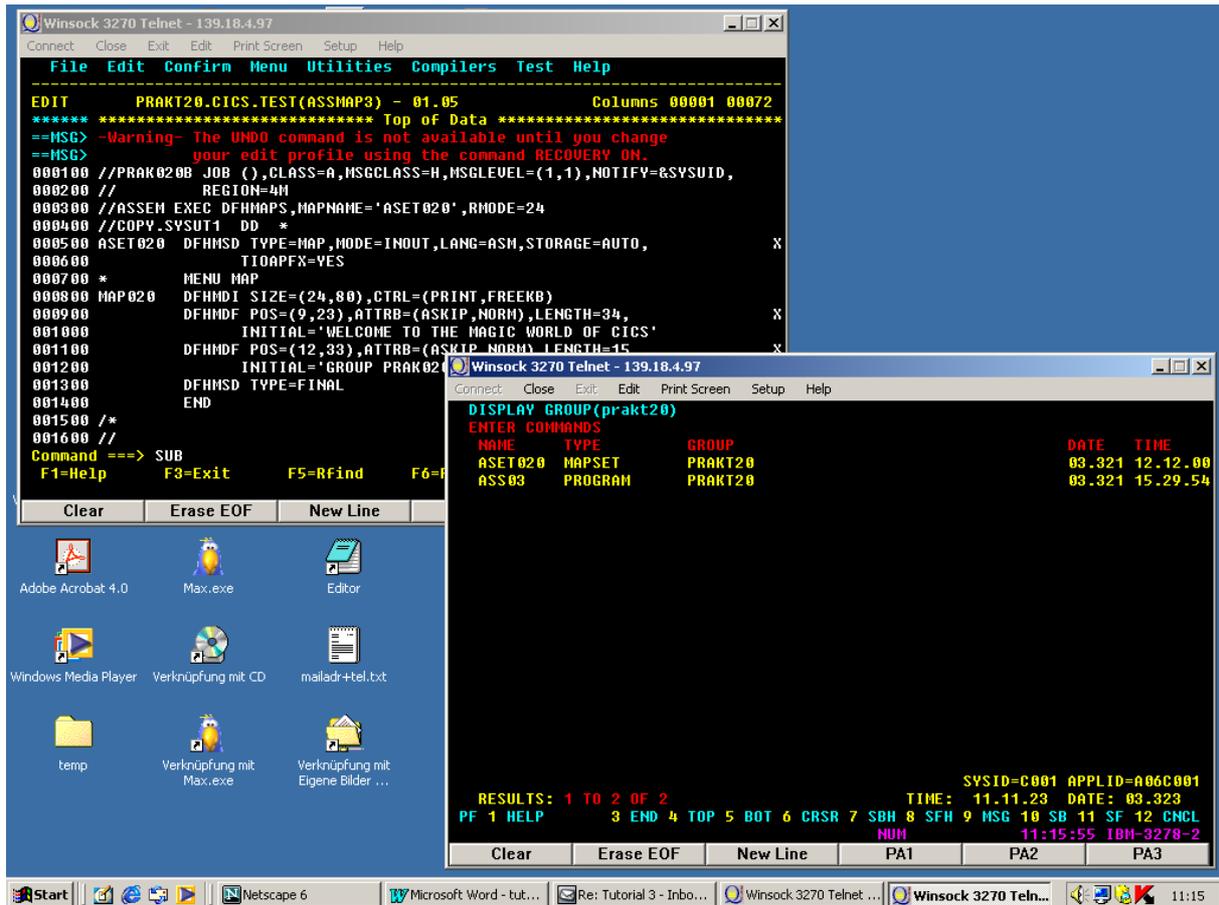


Abbildung 3: eine TSO- sowie eine CICS-Session auf dem Desktop

Wir entwickeln unsere Anwendung unter TSO. Wir wollen sie unter CICS laufen lassen. Dazu muß sie als Teil des CICS-Subsystems installiert werden. Es ist komfortabel, mit 2 z/OS-Sessions gleichzeitig zu arbeiten.

```

z/OS Z18 Level 0609                                IP Address = 91.67.197.244
                                                    VTAM Terminal = SCOTCP49

Application Developer System

          // 0000000 SSSSS
         // 00 00 SS
zzzzzz // 00 00 SS
        zz // 00 00 SSSS
       zz // 00 00 SS
      zz // 00 00 SS
zzzzzz // 0000000 SSSS

System Customization - ADCD.Z18.*

===> Enter "LOGON" followed by the TSO userid. Example "LOGON IBMUSER" or
===> Enter L followed by the APPLID
===> Examples: "L TSO", "L CICS", "L IMS3270

l tso
    
```

Abbildung 4: Logon-Bildschirm

Wir loggen uns anstatt mit "l tso" mittels "L CICS" ein (einschließlich Eingabetaste) und rufen damit das CICS-Subsystem auf (s. Abbildung 4).

```

                          Signon to CICS                                APPLID A06C001

----- WELCOME AT UNIVERSITY OF LEIPZIG -----          -JEDI-
BITTE TRANSAKTION <CESF LOGOFF> ZUM AUSLOGGEN BENUTZEN!    -CICS-

Type your userid and password, then press ENTER:

  Userid . . . . PRAKT20      Groupid . . .
  Password . . . *****
  Language . . .

  New Password . . .

DFHCE3520 Please type your userid.
F3=Exit
    
```

Abbildung 5: Signon to CICS-Bildschirm

Der Signon to CICS-Bildschirm erscheint (s. Abbildung 5). Hier müssen wir unseren TSO-Login-Namen sowie das entsprechende Passwort eingeben. Die Eingabetaste führt uns in den nächsten Screen (s. Abbildung 6).



Abbildung 6: Sign-on is complete

Mit der "Tab"-Taste bewegen wir den Cursor auf die unterste Zeile.

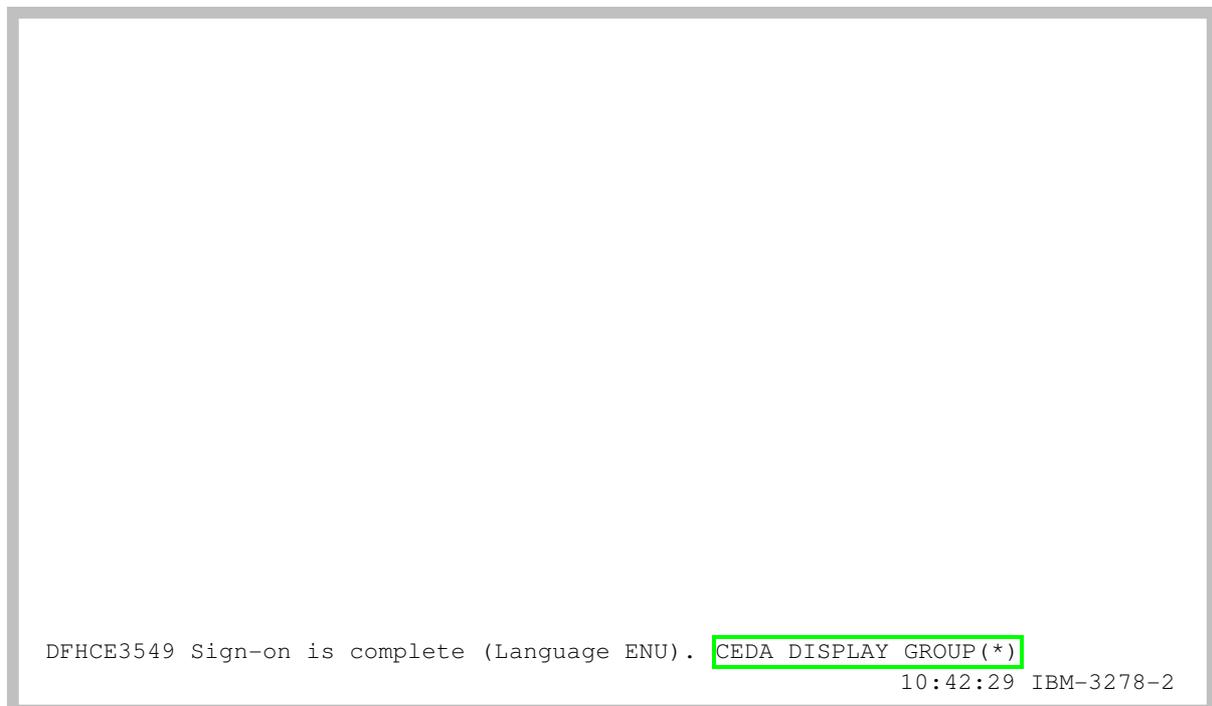


Abbildung 7: Beispielkommando "CEDA DISPLAY GROUP(*)"

CICS erwartet, dass man eine (von vielen) Transaktionen aufruft. Die unterschiedlichen Transaktionen werden normalerweise durch die Eingabe einer aus vier Zeichen bestehenden Transaktions-ID aufgerufen.

Der CICS-Kommandointerpreter ist ebenfalls als Transaktion implementiert. Er wird mit der internen Transaktions-ID "CEDA" aufgerufen, gefolgt von einer Parameterliste, die CICS-Kommandos sowie Eingabedaten enthält.

Als Beispiel geben wir das Kommando "CEDA DISPLAY GROUP(*)" gefolgt von der Eingabetaste ein (s. Abbildung 7).

```

display group(*)
ENTER COMMANDS
  GROUP
  AOR2TOR
  ART
  ATC
  CBPS
  CEE
  CICREXX
  CICSJADP
  CICS0ADP
  CSQ
  CSQCKB
  CSQSAMP
  CTA1TCP
  C001EZA
  C001TCP
  DAVIN15
  DAVIN4
+ DAVIN8

                                     SYSID=C001 APPLID=A06C001
RESULTS: 1 TO 17                               TIME: 00.00.00 DATE: 01.035
PF 1 HELP          3 END 4 TOP 5 BOT 6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Abbildung 8: Auflistung der Gruppen

Wenn unter CICS Anwendungen (Transaktionen) installiert werden, dann wird für jede Transaktion eine "Group" angelegt. In der "Group" befinden sich Members wie das Anwendungsprogramm selbst, der dazugehörige Mapset sowie ein Eintrag, der die Transaktion mit einer 4-stelligen TRID (**TR**ansaktions-**ID**) verknüpft.

Eine Liste der einmal installierten Gruppen ist in Abbildung 8 dargestellt.

```

CEDA DEFINE MAPSET(ASET020) GROUP(PRAKT20)
ENTER COMMANDS
GROUP
AOR2TOR
ARTT
ATC
CBPS
CEE
CICREXX
CSQ
CSQCKB
CSQSAMP
CTAITCP
C001EZA
C001TCP
DBA1
DFH$ACCT
DFH$AFFY
DFH$AFLA
+ DFH$BABR

RESULTS: 1 TO 17
PF 1 HELP          3 END 4 TOP 5 BOT 6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

SYSID=C001 APPLID=A06C001
TIME: 00.00.00 DATE: 01.037

```

Abbildung 9: Definition des Mapsets "ASET020" und der Gruppe "PRAKT20"

Wir definieren für unsere Transaktion eine eigene Gruppe "PRAKT20" und den dazugehörigen Mapset als "ASET020". Hierzu überschreiben wir die oberste Zeile, die als Kommandozeile dient, mit dem CEDA-Befehl "CEDA DEFINE MAPSET(ASET020) GROUP(PRAKT20)" (s. Abbildung 9, bitte Großbuchstaben benutzen!)

Um zu bestätigen, drücken wir die Eingabetaste.

```

CEDA DEFINE MAPSET(ASET020) GROUP(PRAKT20)
OVERTYPE TO MODIFY
CEDA DEFINE Mapset( ASET020  )
Mapset      : ASET020
Group       : PRAKT20
Description ==>
Resident    ==> No                No | Yes
USAge       ==> Normal            Normal | Transient
USElpacopy  ==> No                No | Yes
Status      ==> Enabled           Enabled | Disabled
RSl         : 00                  0-24 | Public

CICS RELEASE = 0530

I New group PRAKT20 created.

SYSID=C001 APPLID=A06C001
TIME: 00.00.00 DATE: 01.037
DEFINE SUCCESSFUL
PF 1 HELP 2 COM 3 END          6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Abbildung 10: Bestätigung der Erstellung der Gruppe "PRAKT20"

CICS teilt uns mit, dass die neue Gruppe "PRAKT20" erstellt und in ihr der Mapset "ASET020" erfolgreich definiert wurde (s. Abbildung 10).

```

CEDA DEFINE PROGRAM(ASSPR020) GROUP(PRAKT20)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFine Mapset( ASET020 )
  Mapset      : ASET020
  Group       : PRAKT20
  Description ==>
  RESident    ==> No                No ! Yes
  USAge       ==> Normal            Normal ! Transient
  USElpacopy  ==> No                No ! Yes
  Status      ==> Enabled            Enabled ! Disabled
  RSl         : 00                  0-24 ! Public

I New group PRAKT20 created.

                                     SYSID=C001 APPLID=A06C001
DEFINE SUCCESSFUL                     TIME: 12.15.03 DATE: 04.068
PF 1 HELP          3 END                6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Abbildung 17: Definition des Programms "ASSPR020"

Wir definieren für die Gruppe "PRAKT20" unser Anwendungsprogramm als "ASSPR020", indem wir den entsprechenden CEDA-Befehl

"CEDA DEFINE PROGRAM(ASSPR020) GROUP(PRAKT20)"

in die oberste Zeile schreiben und anschließend die Eingabetaste betätigen (s. Abbildung 17).

```

CEDA DEFINE PROGRAM(ASSPR020) GROUP(PRAKT20)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
  CEDA DEfIne PRoGAm( PROG020  )
    PRoGAm      : ASSPR020
    GRoUp       : PRAKT20
    DEsCRiption ==>
    LAnguage    ==>                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: 01.037
PF 1 HELP 2 COM 3 END                          6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Abbildung 18: Definition von "ASSPR020"

CEDA will einiges von uns wissen (s. Abbildung 18). Wir übernehmen alle Default-Werte und geben als Sprache "Le370" an (s. Abbildung 19).

```

CEDA DEFINE PROGRAM(PROG020) GROUP(PRAKT20)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
  CEDA DEfIne PRoGAm( ASSPR020  )
    PRoGAm      : ASSPR020
    GRoUp       : PRAKT20
    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: 01.037
PF 1 HELP 2 COM 3 END                          6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Abbildung 19: Auswahl der Sprache

Mit Betätigung der Eingabetaste erscheint der nächste Screen (s. Abbildung 20).

```

OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE PROGRAM( ASSPR020  )
  PROGram      : ASSPR020
  Group        : PRAKT20
  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
  QConcurency  ==> Quasirent            Quasirent | Threadsafe
REMOTE ATTRIBUTES
  DYNAMIC      ==> No                   No | Yes
+ REMOTESystem ==>

                                           SYSID=C001 APPLID=A06C001
  DEFINE SUCCESSFUL                       TIME: 00.00.00 DATE: 01.037
PF 1 HELP 2 COM 3 END                     6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Abbildung 20: Erfolgreiche Definition

Wir verlassen die Definition mit der Eingabe von F3, als Ergebnis davon erscheint "SESSION ENDED" (s. Abbildung 21).

```

CEDA CEDA DEFINE PROGRAM(ASSPR020) GROUP(PRAKT20)
STATUS: SESSION ENDED

```

Abbildung 21: Definition wurde beendet

In diesen Bildschirm geben wir in die oberste Zeile den nächsten CEDA-Befehl ein, gefolgt von der Eingabetaste.

```
CEDA DEFINE TRANS(V020) GROUP(PRAKT20)
STATUS:  SESSION ENDED
```

Abbildung 22: Definition der Transaktion V020

Unsere Transaktion soll wie alle anderen Transaktionen vom Bildschirm über eine 4-stellige Transaktions-ID aufgerufen werden. Wir wählen hierfür die ID "V020" und teilen diese Wahl mit Hilfe des "CEDA DEFINE"-Befehls mit (s. Abbildung 22). Genauso wie "PROG020" wird diese Bestandteil der Gruppe "PRAKT20". Abschließend betätigen wir die Eingabetaste.

```
DEFINE TRANS(V020) GROUP(PRAKT20)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFine TRAnsAction( V020 )
  TRAnsAction ==> V020
  Group       ==> PRAKT20
  DEscription ==>
  PROGram    ==>
  TWasize    ==> 00000                                0-32767
  PROFile    ==> DFHCICST
  PArtitionset ==>
  SStatus    ==> 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
```

Abbildung 23: Der Definitions-Screen

CEDA will mehrere Angaben von uns wissen und schlägt eine Reihe von Default-Werten vor (s. Abbildung 23).

```

DEFINE TRANS(V020) GROUP(PRAKT20)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE TRANSAction( V020 )
  TRANSAction  ==> V020
  Group        ==> PRAKT20
  Description  ==>
  PROGram     ==> ASSPR020
  TWasize     ==> 00000           0-32767
  PROFile     ==> DFHCICST
  PArtitionset ==>
  SStatus     ==> 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

```

Abbildung 24: Eingabe der Parameter

Wir übernehmen alle Default-Werte und geben in die Zeile "PROGram" den Namen unseres Anwendungsprogramms "ASSPR020" ein und bestätigen mit der Eingabetaste (s. Abbildung 24).

```

OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE TRANSAction( V020 )
  TRANSAction  : V020
  Group        : PRAKT20
  Description  ==>
  PROGram     ==> ASSPR020
  TWasize     ==> 00000           0-32767
  PROFile     ==> DFHCICST
  PArtitionset ==>
  SStatus     ==> 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: 22.00.13 DATE: 01.037
PF 1 HELP 2 COM 3 END                    6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Abbildung 25: Erfolgreiche Definition

Die Meldung "DEFINE SUCESSFUL" erscheint (s. Abbildung 25).

Wir verlassen dieses Menü mit F3.

```
CEDA DEFINE TRANS(V020) GROUP(PRAKT20)  
STATUS: SESSION ENDED
```

Abbildung 26: Nach dem Verlassen des Menüs

Der Bildschirm in der Abbildung 26 erscheint. Wir haben CICS den Namen unseres Anwendungsprogramms und eine dazugehörige Transaktions-ID bekanntgegeben. Jetzt müssen diese in die CICS-Programmbibliothek übernommen (installiert) werden.

```
CEDA INSTALL GROUP(PRAKT20)  
STATUS: SESSION ENDED
```

Abbildung 27: Aufruf der Installation

Wir geben in die oberste Zeile das CEDA-INSTALL-Kommando ein und drücken die Eingabetaste (s. Abbildung 27).

```

INSTALL GROUP(PRAKT20)
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: 22.02.15 DATE: 01.037
INSTALL SUCCESSFUL
PF 1 HELP          3 END          6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Abbildung 28: Installation war erfolgreich

CEDA bestätigt, dass die Installation erfolgreich war (s. Abbildung 28). Wir verlassen mit F3 dieses Menü.

```

CEDA INSTALL GROUP(PRAKT20)
STATUS: SESSION ENDED

```

Abbildung 29: Beendete Installation

Der obige Bildschirm erscheint (s. Abbildung 29). Unsere Transaktion ist als Teil der CICS-Anwendungsbibliothek installiert worden und kann nun aufgerufen und damit ausgeführt werden. Dafür löschen wir die oberste Zeile (die CEDA-Kommandozeile) ganz, und rufen unsere Anwendung auf, indem wir dort unsere Transaktions-ID, nämlich "V020", eingeben.



Abbildung 30: Ausgabe der Transaktion "V020" auf dem Bildschirm

Nach dem Betätigen der Eingabetaste erscheint unsere CICS-Transaktion auf dem Monitor (s. Abbildung 30).

Wir betätigen die Eingabetaste, um zum nächsten Screen zu gelangen.

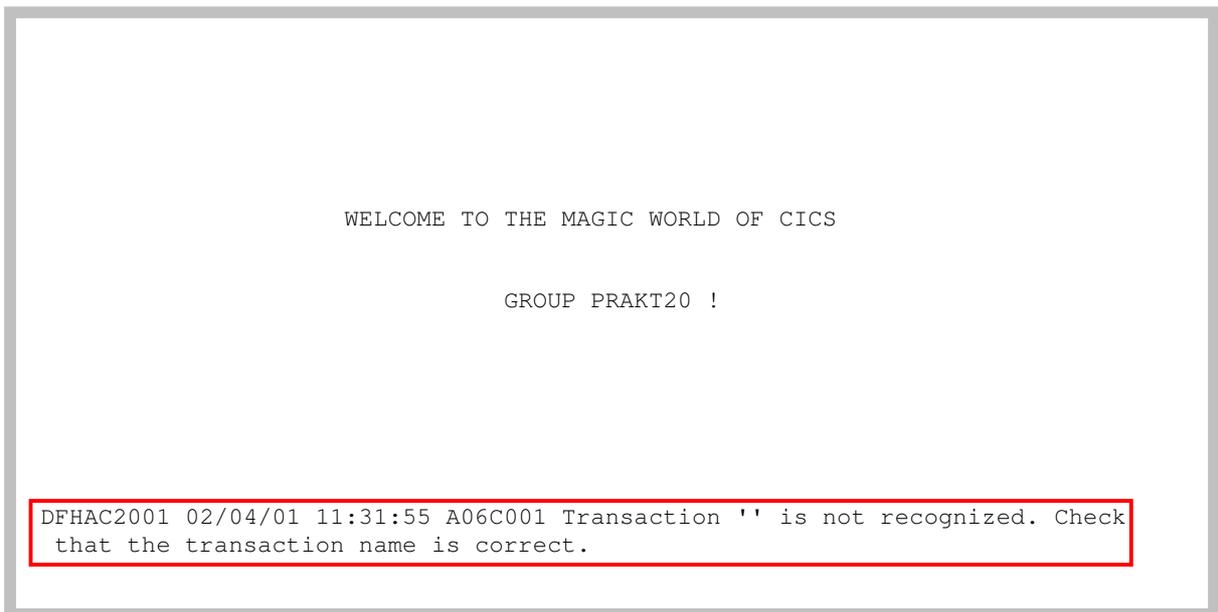


Abbildung 31: Fehlermeldung

Dies terminiert die Bildschirmausgabe unserer Transaktion und erzeugt wieder eine (belanglose) Fehlermeldung (s. Abbildung 31).

```
WELCOME TO THE MAGIC WORLD OF CICS

GROUP PRAKT20 !

DFHAC2001 02/04/01 11:31:55 A06C001 Transaction '' is not recognized. Check
that the transaction name is correct. CEDA DISPLAY GROUP(PRAKT20)
```

Abbildung 32: Befehlseingabe zur Ansicht aller gespeicherter Daten in Group "PRAKT20"

Alle Bestandteile unserer Transaktion sind in der Gruppe PRAKT20 gespeichert. Wir sehen sie uns an, indem wir den Befehl "CEDA DISPLAY GROUP(PRAKT20)" eingeben und anschließend die Eingabetaste drücken (s. Abbildung 32).

```
DISPLAY GROUP(PRAKT20)
ENTER COMMANDS
NAME      TYPE      GROUP      DATE      TIME
ASET020  MAPSET    PRAKT20   01.034   24.00.00
ASSPR020 PROGRAM    PRAKT20   01.034   24.00.00
V020     TRANSACTION PRAKT20   01.034   24.00.00

RESULTS: 1 TO 3 OF 3
PF 1 HELP      3 END 4 TOP 5 BOT 6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

SYSID=C001 APPLID=A06C001
TIME: 00.00.00 DATE: 01.035
```

Abbildung 33: Ausgabe aller Komponenten der Gruppe "PRAKT20"

Die Gruppe "PRAKT20" besteht aus den drei Komponenten "ASET020", "ASSPR020" und "V020", die wir unter CICS definiert und anschließend installiert haben.

Aufgabe: Bereiten Sie die CICS-Transaktion vor und führen Sie diese dann aus. (Hinweis: Die Übersetzung des Assembler-Programms mittels des JCL-Scriptes "ASSSTA03" erfolgt fehlerfrei, wenn diese mit der Statusmeldung "MAXCC=0" beendet wird). Benutzen Sie als CICS-Gruppennamen ihren Login-Namen (z.B. "PRAKT20" oder "PRAK165") und als TRID "V<Ihre Prakt-Nr>". Haben Sie z.B. den Account "PRAKT15, dann ist "V015" Ihre TRID.

Aufgabe: Erzeugen Sie per Print-Screen ein Bild ihres Fensters, welches die Bildschirmausgabe von CICS, ähnlich der Abbildung 2, enthält. Dieses muß unbedingt den **Transaktionsnamen**, "**TUTORIAL 3 IN ASSEMBLER**", das **Bearbeitungsdatum** sowie den **Namen oder die Namen der Autoren** enthalten. Achten Sie darauf, dass das Bild nicht mehr als 250 KByte Speicherplatz belegt. Sehr gut ist das JPEG-Format, das mit weniger als 90 KByte auskommt. Schicken Sie Ihrem Betreuer dieses Bild als Bitmap- oder JPEG-Bild zu. Die Daten ihrer Arbeit dürfen nicht gelöscht werden, so dass Ihr Betreuer Ihre Transaktion aufrufen kann.

Um uns aus CICS auszuloggen, betätigen wir einmal oder mehrmals die Taste F3, bis die Meldung "STATUS: SESSION ENDED" ausgegeben wird. In die Zeile darüber geben wir die Logoff-Transaktion "CESF LOGOFF" ein, gefolgt von der Eingabetaste (s. Abbildung 34).



Abbildung 34: Ausloggen aus CICS