

Tutorial 3

CICS

Copyright © Institut für Informatik, Universität Leipzig

ph v/2010/03

Ziel dieser Aufgabe ist es, ein "Hello World"-Programm zu schreiben und mittels CICS auf dem Bildschirm auszugeben.

Aufgabe: Arbeiten Sie nachfolgendes Tutorial durch.

TSO und CICS sind z/OS-Subsysteme, 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.

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

Wir öffnen den "Data Set Utility"-Screen und erzeugen (Allocate) neue Partitioned Datasets: "PRAKT20.CICS.TEST" und "PRAKT20.LIB". Dabei verwenden wir die in der nachstehenden Aufgabe 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 die Datasets "PRAKT20.CICS.TEST", "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>

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.TEST".

Wir rufen den "Edit Entry Panel Screen" auf und editieren ein Member "BMSPROG" für den neu angelegten Partitioned Dataset "PRAKT20.CICS.TEST" (s. Abbildung 1).

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          PRAKT20.CICS.TEST(BMSPROG) - 01.02          Columns 00001 00072
*****      ***** Top of Data *****
==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 //PRAKT20P JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID
000002 //ASSEM EXEC DFHMAPS,MAPNAME='MSET020',RMODE=24
000003 //SYSUT1 DD *
000004 MSET020 DFHMSD TYPE=MAP,MODE=INOUT,LANG=C,STORAGE=AUTO,TIOAPFX=YES
000005 *      MENU MAP.
000006 MAP020 DFHMDI SIZE=(24,80),CTRL=(PRINT,FREEKB)
000007 DFHMDF POS=(9,23),ATTRB=(ASKIP,NORM),LENGTH=34, X
000008 INITIAL='WELCOME TO THE MAGIC WORLD OF CICS'
000009 DFHMDF POS=(12,33),ATTRB=(ASKIP,NORM),LENGTH=15, X
000010 INITIAL='GROUP PRAKT20 !'
000011 DFHMSD TYPE=FINAL
000012 END
000013 /*
000014 //
Command ==>          Scroll ==> PAGE
F1=Help      F3=Exit      F5=Rfind      F6=Rchange  F12=Cancel

```

Abbildung 1: Das BMS-Programm

Unser BMS-Programm verwendet 3 Befehlstypen: DFHMSD, DFHMDI und DFHMDF.

Ein BMS-Bildschirm verwendet das 24 Zeilen x 80 Zeichen / Zeile 3270-Bildschirmformat. Die Ein- und Ausgabemasken werden als Felder innerhalb der 24x80-Matrix dargestellt, jeweils mit der Angabe: Zeilenadresse, Spaltenadresse und Feldlänge. Dies geschieht mit Hilfe des DFHMDF-Befehls. Der DFHMDF-Befehl in Zeile 000007 definiert ein Feld, welches in Zeile 9, Spalte 23 beginnt, 34 Zeichen lang ist, und mit dem Wert "WELCOME TO THE MAGIC WORLD OF CICS" initialisiert wird.

Der DFHMSD-Befehl (Zeile 000004) definiert einen "Mapset" mit dem Namen "MSET020". Eine Transaktion involviert in der Regel mehrere unterschiedliche Screens, z.B einen Screen, in dem der Benutzer zu einer Eingabe aufgefordert wird und einen weiteren Screen, welcher die Ergebnisse der Anfrage wiedergibt. Alle Maps (Screens) eines Transaktionstyps werden zu einem Mapset zusammengefaßt.

Die einzelnen Maps (Screens) eines Mapsets werden durch den Befehl DFHMDI definiert und zur Kennzeichnung mit einem Namen versehen. In unserem einfachen "Hello World"-Beispiel besteht der Mapset aus einem einzigen Map, der durch die Bezeichnung "MAP020" in Zeile 000006 definiert wird.

Wir geben auf der Kommandozeile den ISPF-Befehl "SUB" ein. Es wird die Prozedur "DFHMAPS" ausgeführt.

JCL findet das Member "DFHMAPS" (Zeile 000002) in der Library "SYS1.PROCLIB(DFHMAPS)". Durch die Ausführung von "DFHMAPS" werden zwei Ausgabe-Files erzeugt. Einmal wird der übersetzte BMS-Quellcode in einen Member in eine

CICS-interne Library mit dem Namen "CICSTS13.CICS.PRAKLOAD" gestellt. Hier kann ihn die BMS-Komponente des CICS-Subsystems später finden.

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
VIEW          PRAKT20.LIB(MSET020) - 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.
000001 union
000002 {
000003 struct {
000004     char          dfhms1Ý12";
000005     } map020i;
000006
000007 struct {
000008     char          dfhms2Ý12";
000009     } map020o;
000010
000011 } map020;
000012
***** ***** Bottom of Data *****

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

Scroll ==> PAGE

```

Abbildung 2: Das Member "MSET020"

Das zweite Ausgabe-File wird als Member "MSET020" in den Partitioned Dataset "PRAKT20.LIB" gestellt (s. Abbildung 2).

Dies ist als Hilfe für die Erstellung des (in C zu schreibenden) Business Logic-Programms gedacht. Alle Ein- und Ausgabe-Masken, die auf dem Bildschirm wiedergegeben werden sollen, werden ja bereits durch das BMS-Programm definiert. Das Business Logic-Programm bearbeitet diese Daten als C-Strukturen, und diese C-Strukturen werden von DFHMAPS während der Übersetzung von "BMSPROG" gleich miterzeugt und in "PRAKT20.LIB(MSET020)" abgespeichert.

Beim Erzeugen des Business Logic-Programms bietet es sich an, "PRAKT20.LIB(MSET020)" als Basis zu benutzen und zu vervollständigen

Aufgabe: Schreiben Sie das BMS-Programm und übersetzen Sie dieses. Modifizieren Sie die beiden Strings "WELCOME TO THE MAGIC ..." und "GROUP ..." so, dass diese Sie als Autor(en) eindeutig identifizieren, z.B. so: "ICH BIN DIE CICS-ANWENDUNG ... VON TEILNEHMER / GRUPPE 4...". (Hinweis: Eine fehlerlose Übersetzung des BMS-Programms wird durch die Statusmeldung "MAXCC=0" beendet). Jeder Teilnehmer (oder jede Gruppe) sollte in CICS einen eigenen Mapset und eine eigene Map nutzen. Ersetzen Sie deshalb im Mapset- sowie Mapnamen "020" durch die Nummer Ihres Prakt-Logins auf Lucas.

Wir erzeugen einen neuen Member "PROG020" des Partitioned Datasets "PRAKT20.CICS.TEST".

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
VIEW      PRAKT20.CICS.TEST(PROG020) - 01.02          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 </'PRAKT20.LIB(MSET020) '>
000002 main()
000003 {
000004     EXEC CICS SEND MAP("map020") MAPSET("mset020") ERASE;
000005 }
*****  ***** Bottom of Data *****

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

Scroll ==>> PAGE

```

Abbildung 3: Das kopierte Member "MSET020"

Mit dem Befehl `#include </'PRAKT20.LIB(MSET020) '>` kopieren wir den vorher erhaltenen "PRAKT20.LIB(MSET020)" in das C-Programm.

Die main-Routine enthält einen einzigen Befehl, einen CICS-Befehl "SEND MAP".

Dieser bewirkt, dass die Map mit der Adresse "MAP020" aus dem Mapset "MSET020" mit Hilfe des 3270-Protokolls an den Bildschirm des Endbenutzers übertragen wird.

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT      PRAKT20.CICS.TEST(START01) - 01.03          Columns 00001 00072
*****  ***** Top of Data *****
==MSG> -CAUTION- Profile changed to NUMBER ON STD (from NUMBER OFF).
==MSG>          Data has 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.
000010 //PRAKT20S JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
000020 //          TIME=1440
001100 //          EXEC PROC=CTOCICS,REG=0M
001300 //TRN.SYSIN DD DISP=SHR,DSN=PRAKT20.CICS.TEST(PROG020)
001400 //LKED.SYSIN DD *
001500          NAME PROG020(R)
*****  ***** Bottom of Data *****

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

Scroll ==>> PAGE

```

Abbildung 4: JCL-File "START01"

Dieses Programm soll nun übersetzt werden. Dazu wird für den Partitioned Dataset "PRAKT20.CICS.TEST" ein neues Member "START01" erstellt (s. Abbildung 4). Dies ist ein JCL-File, das die Prozedur CTOCICS (Compile to CICS) enthält. CTOCICS ruft zunächst den CICS-Precompiler auf, der alle CICS-Befehle in C-Befehle übersetzt. Anschließend wird der C-Compiler aufgerufen, der ein Maschinenprogramm erstellt und in eine für das CICS-Subsystem zugängliche Library stellt.

Wir geben "SUB" ein und warten, bis der Job ausgeführt wurde (s. Abbildung 4).

Die Anwendung ist jetzt unter TSO entwickelt und kann nun im CICS-Subsystems installiert werden. Dazu loggen wir uns unter TSO aus und unter CICS ein.

```

z/OS Z18 Level 0609                                IP Address = 91.67.197.24
                                                    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 5: Logon-Bildschirm

Wir loggen uns anstatt "l tso" mit "L CICS" ein(einschließlich Eingabetaste) und rufen damit das CICS-Subsystem auf (s. Abbildung 5)

Abbildung 5

```
                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 6: Signon to CICS-Bildschirm

Der Signon to CICS-Bildschirm erscheint (s. Abbildung 6). Hier müssen wir unseren TSO-Loginnamen sowie das entsprechende Passwort eingeben. Die Eingabetaste führt uns in den nächsten Screen (s. Abbildung 7).

```
DFHCE3549 Sign-on is complete (Language ENU).
                                           10:41:29 IBM-3278-2
```

Abbildung 7: Sign-on is complete

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

```

DFHCE3549 Sign-on is complete (Language ENU).  CEDA DISPLAY GROUP (*)
                                                    10:42:29 IBM-3278-2

```

Abbildung 8: 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, welche CICS-Kommandos sowie Eingabedaten enthält.

Als Beispiel geben wir das Kommando "CEDA DISPLAY GROUP(*)" gefolgt von der Eingabetaste ein (s. Abbildung 8).

```

display group(*)
ENTER COMMANDS
  GROUP
  AOR2TOR
  ARTT
  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 9: Auflistung der Gruppen

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

Es werden die ersten 17 der installierten Gruppen angezeigt (s. Abbildung 9). Weitere Gruppen werden durch Blättern (Taste F8) sichtbar.

***Aufgabe:** Falls noch eine alte CICS-Gruppe mit dem Namen Ihrer User-ID installiert ist, löschen Sie diese. Dies ist möglich mittels `CEDA DELETE ALL GROUP(<Ihre User-ID>)`. Sie müssen das tun, damit - wie folgend beschrieben - Sie eine neue CICS-Gruppe einschließlich ihrer drei Komponenten Mapset, Programm und Transaktions-ID anlegen können.*

```

DEFINE MAPSET (MSET020) GROUP (PRAKT20)
ENTER COMMANDS
GROUP
AOR2TOR
ARTT
ATC
CBPS
CEE
CICREXX
CSQ
CSQCKB
CSQSAMP
CTA1TCP
C001EZA
C001TCP
DBA1
DFH$ACCT
DFH$AFFY
DFH$AFLA
+ DFH$BABR

                                SYSID=C001 APPLID=A06C001
RESULTS: 1 TO 17                TIME: 00.00.00 DATE: 01.037
PF 1 HELP          3 END 4 TOP 5 BOT 6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Abbildung 10: Definition des Mapsets "MSET020" und der Gruppe "PRAKT20"

Wir definieren für unsere Transaktion eine eigene Gruppe "PRAKT20" und den dazugehörigen Mapset als "MSET020". Hierzu überschreiben wir die oberste Zeile, die als Kommandozeile dient, mit dem CEDA-Befehl "DEFINE MAPSET(MSET020) GROUP(PRAKT20)" (s. Abbildung 10, bitte Großbuchstaben benutzen!)

Um zu bestätigen drücken wir die Eingabetaste.


```

CEDA DEFINE MAPSET(MSET020) GROUP(PRAKT20)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFine Mapset( MSET020  )
  Mapset      : MSET020
  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.

DEFINe SUCCESSFUL                                SYSID=C001 APPLID=A06C001
                                                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 11: Bestätigung der Erstellung der Gruppe "PRAKT20"

CICS teilt uns mit, dass die neue Gruppe "PRAKT20" erstellt wurde und dass der Mapset erfolgreich definiert wurde (s. Abbildung 11).

Führen wir nochmals den Befehl "CEDA DISPLAY GROUP(*)" aus, so finden wir in der dargestellten Liste den Eintrag "PRAKT20" (erfordert ein Durchblättern der Liste unter Nutzung der Taste F8).

```

CEDA DEFINE PROGRAM(PROG020) GROUP(PRAKT20)
STATUS:  SESSION ENDED

```

Abbildung 12: Definition des Programms "PROG020"

Wir betätigen ein- oder mehrfach die F3-Taste, bis "STATUS: SESSION ENDED" erscheint.

Anschließend definieren wir für die Gruppe "PRAKT20" unser Anwendungsprogramm als "PROG020", indem wir den entsprechenden CEDA-Befehl

"CEDA DEFINE PROGRAM(PROG020) GROUP(PRAKT20)"

in die oberste Zeile schreiben und anschließend die Eingabetaste betätigen (s. Abbildung 12).

```

DEFINE PROGRAM(PROG020) GROUP(PRAKT20)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFine PROGram( PROG020 )
  PROGram      : PROG020
  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 13: Definition von "PROG020"

CEDA will einiges von uns wissen (s. Abbildung 13). Wir übernehmen alle Default-Werte und geben als Sprache "Le370" an (s. Abbildung 14).

```

DEFINE PROGRAM(PROG020) GROUP(PRAKT20)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFine PROGram( PROG020 )
  PROGram      : PROG020
  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 14: Auswahl der Sprache

"Le370" ist keine Sprache sondern eine Laufzeitumgebung. CICS braucht an dieser Stelle in Wirklichkeit nicht die Angabe der Quellsprache unseres Anwendungsprogramms (wir haben es ja bereits übersetzt), sondern die Angabe der Laufzeitumgebung des von uns verwendeten Compilers. Alle modernen z/OS-Compiler verwenden eine gemeinsame Laufzeitumgebung, die den Namen "Le370" trägt. Mit Betätigung der Eingabetaste wird "Le370" gespeichert und "DEFINE SUCCESSFUL" wird ausgegeben (s. Abbildung 15).

```

OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFINE PROGram( PROG020  )
  PROGram      : PROG020
  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 15: Erfolgreiche Definition

Wir verlassen die Definition des Programms mit der Eingabe von F3, als Ergebnis davon erscheint "SESSION ENDED" (s. Abbildung 16).

```

CEDA DEFINE PROGRAM(PROG020) GROUP(PRAKT20)
STATUS: SESSION ENDED

```

Abbildung 16: Definition wurde beendet

In diesen Bildschirm geben wir in die oberste Zeile den nächsten CEDA-Befehl ein (s. Abbildung 17).

```
CEDA DEFINE TRANS(T020) GROUP(PRAKT20)
STATUS:  SESSION ENDED
```

Abbildung 17: Definition der Transaktion T020

Unsere Transaktion soll wie alle anderen Transaktionen vom Bildschirm über eine 4-stellige Transaktions-ID aufgerufen werden. Wir wählen hierfür die ID "T020" und teilen diese Wahl mit Hilfe des "CEDA DEFINE"-Befehls mit (s. Abbildung 17). Genauso wie das Programm "PROG020" wird diese Bestandteil der Gruppe "PRAKT20". Abschließend betätigen wir die Eingabetaste.

```
DEFINE TRANS(T020) GROUP(PRAKT20)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFine TRANSAction( T020 )
TRANSAction ==> T020
Group        ==> PRAKT20
Description  ==>
PROG         ==>
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
SHuttdown    ==> 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 18: Der Definitions-Screen

CEDA will mehrere Angaben von uns und schlägt eine Reihe von Default-Werten vor (s. Abbildung 18).

```

DEFINE TRANS(T020) GROUP(PRAKT20)
OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFine TRANSAction( T020 )
  TRANSAction ==> T020
  Group        ==> PRAKT20
  Description   ==>
  PROGram      ==> PROG020
  TWasize      ==> 00000                0-32767
  PROFile      ==> DFHCICST
  PArtitionset ==>
  SStatus      ==> Enabled              Enabled | Disabled
  PRIMedsize   : 00000                  0-65520
  TASKDATALoc  ==> Below                Below | Any
  TASKDATAKey  ==> User                  User | Cics
  SOrageclear  ==> 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 19: Eingabe der Parameter

Wir übernehmen alle Default-Werte und geben in die Zeile "PROGram" den Namen unseres Anwendungsprogramms, nämlich "PROG020", ein und bestätigen mit der Eingabetaste (s. Abbildung 19).

```

OVERTYPE TO MODIFY                                CICS RELEASE = 0530
CEDA DEFine TRANSAction( T020 )
  TRANSAction : T020
  Group        : PRAKT20
  Description   ==>
  PROGram      ==> PROG020
  TWasize      ==> 00000                0-32767
  PROFile      ==> DFHCICST
  PArtitionset ==>
  SStatus      ==> Enabled              Enabled | Disabled
  PRIMedsize   : 00000                  0-65520
  TASKDATALoc  ==> Below                Below | Any
  TASKDATAKey  ==> User                  User | Cics
  SOrageclear  ==> 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 20: Erfolgreiche Definition

Die Meldung "DEFINE SUCESSFUL" erscheint (s. Abbildung 20).

Wir verlassen dieses Menü mit F3.

```
CEDA DEFINE TRANS (T020) GROUP (PRAKT20)
STATUS:  SESSION ENDED
```

Abbildung 21: Nach dem Verlassen des Menüs

Der Bildschirm in der Abbildung 21 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 22: Aufruf der Installation

Wir geben in die oberste Zeile das CEDA-INSTALL-Kommando ein und drücken die Eingabetaste (s. Abbildung 22).

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

Abbildung 23: Installation war erfolgreich

CEDA bestätigt, dass die Installation erfolgreich war (s. Abbildung 23). Wir verlassen mit F3 dieses Menü.

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

Abbildung 24: Beendete Installation

Der in Abbildung 24 dargestellte Bildschirm erscheint. 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 "T020", eingeben.

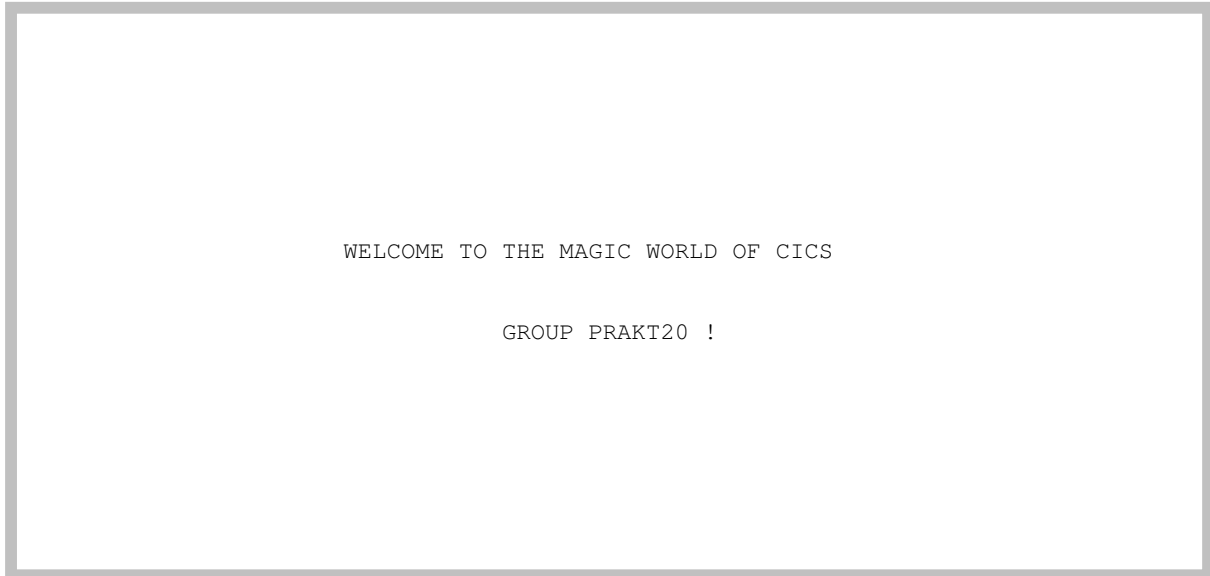


Abbildung 25: Ausgabe der Transaktion "T020" auf dem Bildschirm

Nach dem Betätigen der Eingabetaste erscheint unsere CICS-Transaktion auf dem Monitor (s. Abbildung 25).

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

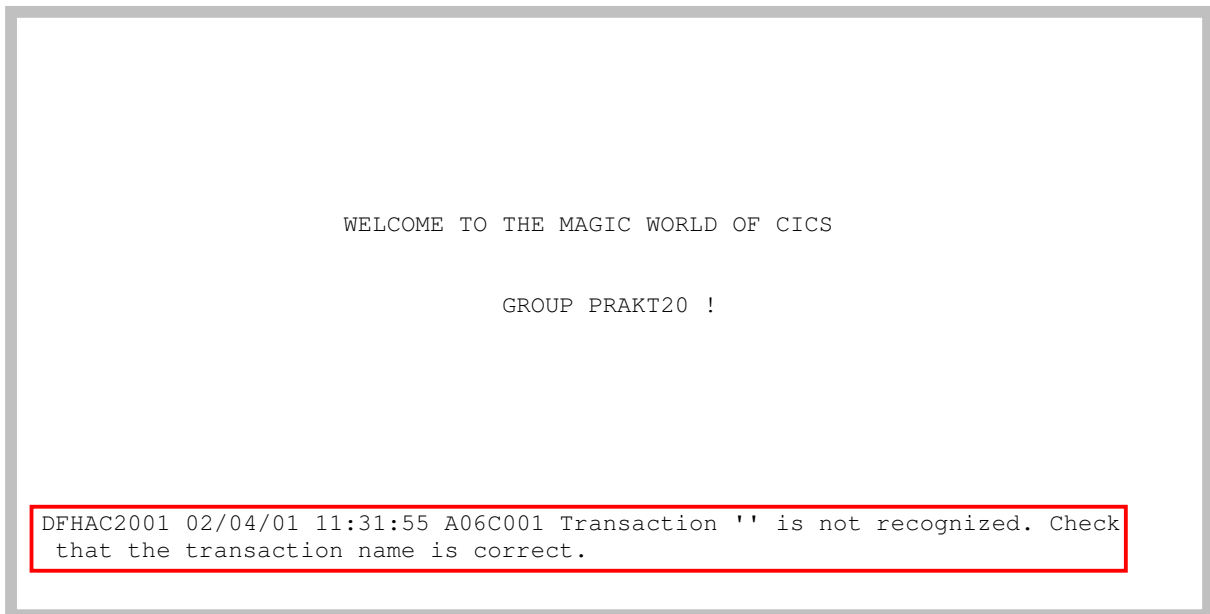


Abbildung 26: Fehlermeldung

Dieser terminiert die Bildschirmausgabe unserer Transaktion und erzeugt wieder eine Fehlermeldung (s. Abbildung 26).


```
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 27: 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 27).

```
DISPLAY GROUP(PRAKT20)
ENTER COMMANDS
NAME      TYPE      GROUP      DATE      TIME
MSET020  MAPSET   PRAKT20   01.034    24.00.00
PROG020  PROGRAM  PRAKT20   01.034    24.00.00
T020     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 28: Ausgabe aller Komponenten der Gruppe "PRAKT20"

Die Gruppe "PRAKT20" besteht aus den drei Komponenten "MSET020", "PROG020" und "T020", 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 C-Programms mittels des JCL-Scriptes "start01" erfolgt fehlerfrei, wenn diese mit der Statusmeldung "MAXCC=4" beendet wird). . Ihr C-Programm speichern Sie unter "PROG<Ihre Prakt-Nr>" ab, so dass jeder Teilnehmer/jede Gruppe sein/ihr eigenes C-Programm in CICS installieren kann. Benutzen Sie als CICS-Gruppennamen ihren Login-Namen (z.B. "PRAKT20" oder "PRAK165") und als TRID "T<Ihre Prakt-Nr>". Haben Sie z.B. den Account "PRAKT15, dann ist "T015" Ihre TRID Erzeugen Sie per Print-Screen ein Bild ihres Fensters, welches die Bildschirmausgabe von CICS enthält (Ihre Version von "WELCOME TO THE MAGIC...", die Sie eindeutig identifiziert). 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 löschen Sie nicht.

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", gefolgt von der Eingabetaste, ein (s. Abbildung 29).



Abbildung 29: Ausloggen aus CICS