

# Tutorial 14

## CICS Transaction Gateway

Copyright © Abt. Computersysteme, Institut für Informatik, Universität Leipzig

Ziel dieser Aufgabe ist es, ein CICS-Programm zu schreiben, das mit dem CICS Transaction Gateway für das Web verfügbar gemacht werden soll.

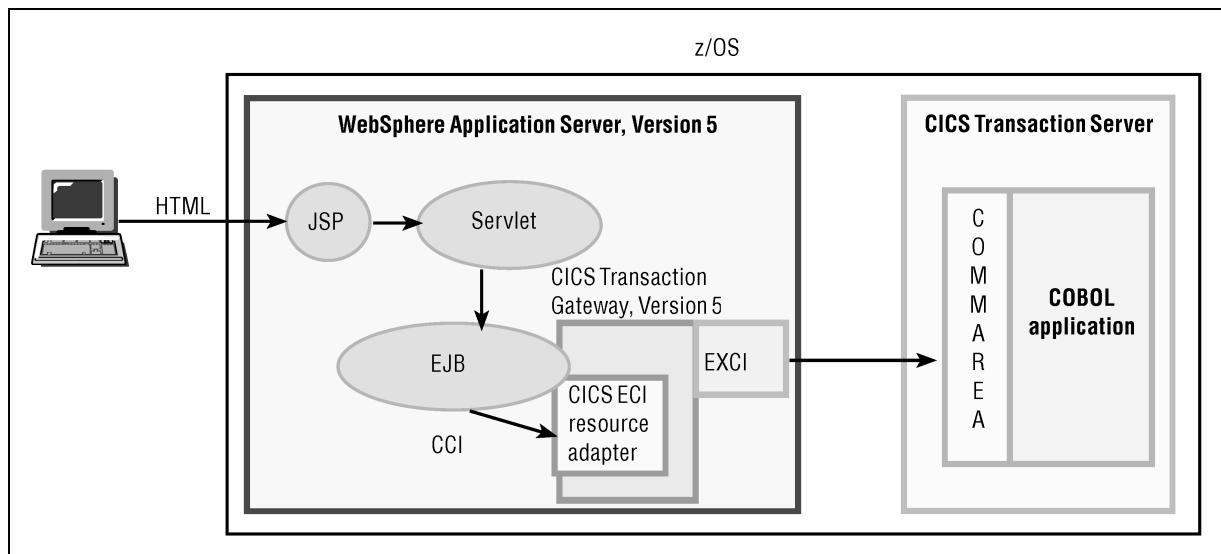


Abbildung 1: Allgemeiner Aufbau

Das CICS Transaction Gateway ist ein recht vielseitiges Stück Software, das mehrere Möglichkeiten zum Zugriff auf CICS-Programme ermöglicht. Wir verwenden hier jedoch nur die Java-Klassen, die auch **Ressourcenadapter** genannt werden. Dieser ist fest im WebSphere Application Server installiert worden und kann von allen J2EE-Anwendungen verwendet werden. Dieses Konzept ähnelt sehr stark JDBC, wobei hier nicht auf Datenbanken, sondern auf CICS-Programme zugegriffen wird.

Im Gegensatz zu J2SE (Java 2 Standard Edition) unterstützt J2EE (Java 2 Enterprise Edition) auch serverseitige Komponenten: Servlets, JSPs und Enterprise JavaBeans. Dabei sind **Servlets** Java-Klassen, die HTTP-Anfragen entgegennehmen und als Antwort dynamisch HTML-Daten zurückliefern. **JSPs** (Java Server Pages) ähneln in der Syntax HTML-Dateien, beinhalten jedoch zusätzliche Elemente, die dynamische Inhalte ermöglichen. **Enterprise JavaBeans** sind serverseitige Java-Klassen, die bestimmten Anforderungen genügen müssen. Sie enthalten die Businesslogik der J2EE-Anwendung, während Servlets und JSPs lediglich für die Präsentationslogik verantwortlich sind.

Der in diesem Tutorial von uns verwendete "CTGTesterCCI" ist unter anderem im IBM-Redbook "CICS Transaction Gateway V5 - The WebSphere Connector for CICS" beschrieben, welches auf der IBM-Webseite frei verfügbar ist.

## Voraussetzung

In diesem Tutorial taucht an vielen Stellen die Nummer "20" oder "020" auf. Bitte ersetzen Sie diese durch Ihre eigene Praktikumsnummer. Angenommen Ihre Nummer ist die 38, dann ersetzen Sie zum Beispiel "PRAKT20" durch "PRAKT38" oder "CTG020" durch "CTG038".

## Überblick

Dieses Tutorial ist für den Uni-Leipzig-Mainframe "Padme" vorgesehen, auf dem unter anderem z/OS 1.5 installiert ist.

Zuerst erstellen und installieren wir ein CICS-Programm namens "CTG20", welches keine Daten auf dem Bildschirm ausgibt, sondern einen Namen in die so genannte **COMMAREA** (Communications Area) schreibt.

Die Funktionsfähigkeit des erstellten und installierten CICS-Programms wird danach mit einem Browser getestet. Der vom CICS-Programm in die COMMAREA geschriebene Name erscheint dabei auf dem Bildschirm, was die Verbindung zwischen CICS und dem Web demonstriert.

## Erstellung des CICS-Programms

Zuerst soll das CICS Programm erstellt werden, auf das das CICS Transaction Gateway später zugreifen soll. Der Unterschied zu den bisherigen CICS-Programmen aus den **Tutorials 3 und 5** besteht darin, dass die Ausgabe nicht in Form von BMS-Maps auf dem Bildschirm erscheint. Stattdessen wird die Ausgabe in die so genannte **COMMAREA** (Communications Area) geschrieben, die zum Datenaustausch zwischen CICS-Programm und CICS Transaction Gateway verwendet wird (Abbildung 1). Solch ein CICS-Programm muss jedoch in COBOL geschrieben werden; es gibt keine Möglichkeit, dieselbe Funktionalität in C bereitzustellen.

Um das CICS-Programm zu erstellen, arbeiten wir sowohl mit dem TSO- als auch mit dem CICS-Subsystem. Wir starten den 3270-Emulator und verwenden dabei die folgenden Parameter.

Parameter	
IP-Adresse:	139.18.4.35
Port:	23
Codepage:	1140 (oder eine andere geeignete Page)

Wir verwenden am Anfang TSO, so dass wir mit dem Befehl "L TSO" eine TSO-Sitzung starten (Abbildung 2) und uns mit unseren Benutzerdaten anmelden.

```

z/OS V1R5 Level 0403                               IP Address = 139.18.4.47
                                                    VTAM Terminal =

Application Developer System

          // 0000000 SSSSSS
          // 00 00 SS
zzzzzz // 00 00 SS
      zz // 00 00 SSSS
      zz // 00 00  SS
      zz // 00 00  SS
zzzzzz // 0000000 SSSSSS

UNI LEIPZIG           - padme.informatik.uni-leipzig.de

===> 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 2: Start der TSO-Sitzung**

Zuerst muss noch ein Dataset erstellt werden. Wir öffnen den "Data Set Utility" Screen und erstellen (Allocate) einen neuen Partitioned Dataset namens PRAKT20.CICS.CTG. Dabei verwenden wir die in der nachstehenden Aufgabe angegebenen Parameter.

**Aufgabe:** *Legen Sie den Dataset "PRAKT20.CICS.CTG" an. Verwenden Sie dazu folgende Parameter:*

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

Unsere Anwendung besteht aus einem Cobol-Programm und einem JCL-Skript zur Übersetzung. Wir erstellen diese im neuen Partitioned Dataset "PRAKT20.CICS.CTG".

In diesem Fall enthält das CICS-Programm lediglich die Businesslogik. Die Präsentationslogik wird später vom WebSphere Application Server in Form einer Webanwendung bereitgestellt.

Wir beginnen mit dem Quellcode des CICS-Programms und editieren ein Member "CTG20" für den neu angelegten Dataset "PRAKT20.CICS.CTG" (Abbildung 3 und Abbildung 4).

Im Quelltext fällt besonders die DFHCOMMAREA in der "LINKAGE SECTION" auf. Diese dient zum Datenaustausch zwischen dem CICS-Programm und dem CICS Transaction Gateway.

**Aufgabe:** Ersetzen Sie im Cobol-Programm in der WORKING-STORAGE SECTION beide Namen durch Ihren jeweiligen Vor- und Zunamen. Übersetzen Sie das Cobol-Programm anschließend so, wie weiter unten beschrieben.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
VIEW          PRAKT20.CICS.CTG(CTG20) - 01.03          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      IDENTIFICATION DIVISION.
000200      PROGRAM-ID. CTG20.
000300      ENVIRONMENT DIVISION.
000400      DATA DIVISION.
000500      WORKING-STORAGE SECTION.
000600      01 NAME-TAB.
000700          02 VORNAME          PICTURE X(20) VALUE "HANS".
000800          02 NACHNAME        PICTURE X(20) VALUE "MEIER".
000900      LINKAGE SECTION.
001000      01 DFHCOMMAREA.
001100          03 VNAM            PICTURE X(20).
001200          03 NNAM            PICTURE X(20).
001300      PROCEDURE DIVISION.
001400          MOVE SPACES TO DFHCOMMAREA.
001500          MOVE VORNAME TO VNAM.
Command ==>          Scroll ==> HALF
F1=Help    F2=Split    F3=Exit    F5=Rfind    F6=Rchange    F7=Up
F8=Down    F9=Swap     F10=Left   F11=Right   F12=Cancel

```

**Abbildung 3: Quelltext des CICS-Programms, Panel 1/2**

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
VIEW          PRAKT20.CICS.CTG(CTG20) - 01.03          Columns 00001 00072
001600          MOVE NACHNAME TO NNAM.
001700          EXEC CICS RETURN
001800              END-EXEC.
001900          EXIT.
*****      ***** 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

```

**Abbildung 4: Quelltext des CICS-Programms, Panel 2/2**

Durch Betätigung der Taste "F3" wird der ISPF-Editor verlassen und das Member automatisch gespeichert. Zur Übersetzung verwenden wir ein JCL-Programm (Abbildung 5). Dieses ruft die Prozedur "DFHYITVL" auf, die alle Übersetzungsschritte für das CICS-Programm enthält: CICS-Übersetzer, Cobol-Compiler und Linker. Wir editieren dazu ein Member namens "STARTEC" des Datasets "PRAKT20.CICS.CTG" (Abbildung 5).

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
VIEW          PRAKT11.CICS.CTG(STARTEC) - 01.04          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 //PRAKT20C JOB  ( ),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID
000200 //S1          EXEC DFHYITVL
000300 //TRN.SYSIN   DD DSN=PRAKT20.CICS.CTG(CTG20),DISP=SHR
000400 //LKED.SYSLMOD DD DSN=CICSTS23.CICS.PRAKLOAD
000500 //LKED.SYSIN   DD *
000600          NAME CTG20(R)
000700 /*
000800 //
***** ***** 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

```

Abbildung 5: JCL-Programm zum Übersetzen des CICS-Programms

Wir geben auf der Kommandozeile den Befehl "SUB" ein (Abbildung 5). Nach der Meldung "IKJ56250I JOB PRAKT20C(JOB07410) SUBMITTED" betätigen wir die Eingabe-Taste. Dann warten wir auf das Ende der Jobverarbeitung (Abbildung 6).

```

14.11.21 JOB07408 $HASP165 PRAKT11C ENDED AT N1  MAXCC=0 CN(INTERNAL)
***

```

Abbildung 6: Bestätigung der Jobverarbeitung

"MAXCC=0" heißt, dass der Job erfolgreich ausgeführt wurde. Das CICS-Programm wurde nun übersetzt und liegt in ausführbarer Form vor.

Nun muss das übersetzte Programm in CICS eingebunden (definiert und installiert) werden. Dazu loggen wird uns im Anmeldebildschirm mittels "L CICS" ein (Abbildung 7).

```
z/OS V1R5 Level 0403                               IP Address = 139.18.4.47
                                                    VTAM Terminal =

Application Developer System

          // 0000000 SSSSSS
          // 00 00 SS
zzzzzz // 00 00 SS
      zz // 00 00 SSSS
      zz // 00 00 SS
zz     // 00 00 SS
zzzzzz // 0000000 SSSSSS

UNI LEIPZIG          - padme.informatik.uni-leipzig.de

===> 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 CICS
```

Abbildung 7: Einloggen ins CICS

Nachdem wir unter unserem Mainframe-Login eingeloggt sind, löschen wir noch durch Betätigung der (eventuell emulierten) CLEAR-Taste das CICS-Fenster. Nun steht der Cursor in der linken oberen Ecke und wartet auf die Eingabe einer CICS-Transaktion.

Wir definieren das Programm mit Hilfe des Befehls "CEDA DEFINE PROGRAM(CTG20) GROUP(PRAKT20)" (Abbildung 8).

```
CEDA DEFINE PROGRAM(CTG20) GROUP(PRAKT20)
```

Abbildung 8: Definition des Programms

CICS möchte von uns nun einige Parameter für das Programm wissen. Wir ändern lediglich die Sprache in "Le370" und bestätigen mit der Eingabetaste (Abbildung 9).

```

OVERTYPE TO MODIFY                                CICS RELEASE = 0630
CEDA DEFINE PROGRAM( CTG11      )
  PROGRAM      : CTG11
  Group        : PRAKT11
  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
  EXEKey       ==> User                  User | Cics
  COncurrency  ==> Quasirent             Quasirent | Threadsafe
  REMOTE ATTRIBUTES
  Dynamic      ==> No                   No | Yes
+ REMOTESystem ==>

                                           SYSID=CICS APPLID=CICS
  DEFINE SUCCESSFUL                       TIME: 14.38.24 DATE: 06.108
PF 1 HELP 2 COM 3 END                     6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Abbildung 9: Parameter des CICS-Programms

Wir installieren daraufhin die Gruppe "PRAKT20" neu, indem wir den CEDA-Befehl "INSTALL GROUP(PRAKT20)" eingeben (Abbildung 10).

```

INSTALL GROUP(PRAKT20)
OVERTYPE TO MODIFY
CEDA Install
  All
  CONNecTion  ==>
  CORbaserver ==>
  DB2Conn     ==>
  DB2Entry    ==>
  DB2Tran     ==>
  DJar        ==>
  DOctemplate ==>
  Enqmodel    ==>
  File        ==>
  Journalmodel ==>
  LSrpool     ==>
  Mapset      ==>
  PARTitionset ==>
  PARTner     ==>
  PROCesstype ==>
+ PROfile     ==>

                                           SYSID=CICS APPLID=CICS
  INSTALL SUCCESSFUL                       TIME: 14.43.51 DATE: 06.108
PF 1 HELP          3 END                     6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Abbildung 10: Installation der Gruppe

## Test der Konfiguration

Nachdem nun das CICS-Programm kompiliert und in CICS definiert und installiert wurde, möchten wir den Aufruf von diesem über das CICS Transaction Gateway testen. Der Test erfolgt in einem normalen Webbrowser Ihrer Wahl.

In den Webbrowser ist als Adresse

<http://padme.informatik.uni-leipzig.de:9148/CTGTesterCCIWeb/>

einzugeben. So wird der CTG-Tester "CTGTesterCCI" gestartet. Nun müssen einige Parameter eingegeben bzw. geändert werden und per Klick auf den Submit-Button wird der Test des Zugriffs auf das CICS-Programm über das CICS Transaction Gateway gestartet.

Aufgabe: *Starten Sie den CTG-Tester "CTGTesterCCI".*

Aufgabe: *Überlegen Sie sich, welche Parameter Sie eventuell ändern müssen und tun Sie das. **Achtung:** Als Wert für "COMMAREA length" muss ein Wert größer Null eingegeben werden. Ein Wert Null ist strengstens verboten, weil dies CICS zum Absturz bringen könnte, was eine Bearbeitung dieses Tutorials weder für Sie noch für andere bis auf weiteres unmöglich machen würde. Klicken Sie auf den SUBMIT-Button. Wenn Ihr Test erfolgreich war, erscheint Ihr Vor- und Zuname mit gelbem Hintergrund auf Ihrem Bildschirm.*

Aufgabe: *Nachdem Ihnen ein erfolgreicher Test gelungen ist, weisen Sie dies bitte anhand eines Screenshots nach (JPEG-Format, max. Größe 99 KB). Dieser soll den gesamten Inhalt des Browser-Fensters zeigen, insbesondere aber Ihren Vor- und Zunamen auf gelbem Untergrund.*

Aufgabe: *Schicken Sie diesen Screenshot im JPEG-Format an Ihren Betreuer*

Aufgabe: *Loggen Sie sich sauber aus dem CICS aus. Dies geschieht mit dem Befehl "CESF LOGOFF". Details dazu sind im Tutorial 3, C-Version, beschrieben. Das Schließen des 3270-Emulator-Fensters ohne ein solches sauberes Ausloggen aus CICS ist ebenfalls verboten, weil dies auch CICS zum Absturz bringen könnte, was ein Bearbeiten der CICS-Tutorien für Sie und andere bis auf weiteres unmöglich machen würde.*