JCICS Tutorial Teil 1
CICS Integration mit Java

Dank an Herrn Stefan Huster für die Bereitstellung des Tutorials

Übersicht

1. Voraussetzungen, Einrichtung und Verwendung der Systemumgebung

In diesem Abschnitt wird die verwendete Entwicklungsumgebung besprochen. Es wird Ihnen gezeigt welche Software Sie für die Bearbeitung dieser Tutorials benötigen und welche Einstellungen Sie innerhalb dieser Programme vornehmen müssen. Am Ende dieses Abschnitts finden Sie eine Anleitung über die Einrichtung der Entwicklungsumgebung und dem An- und Abmeldeprozess unter CICS.

1.1. Verwendete Software

Die in diesem Abschnitt aufgelistete Software muss für eine erfolgreiche Bearbeitung dieser Tutorials auf Ihrem Computer installiert sein.

1.1.1 Java Development Kit

Grundvoraussetzung für die Entwicklung von Java-Anwendungen ist das Java Development Kit (JDK). Dieses enthält den notwendigen Compiler und eine entsprechende Laufzeitumgebung. Für Windows und Linux ist dieses auf der Webseite http://www.java.com/de/download/ zum kostenlosen Download erhältlich. Nutzer von MacOs benötigen die von Apple bereitgestellte Implementierung des JDK. Diese kann im Apple-Entwicklerbereich unter http://developer.apple.com/ gefunden werden. Dieses Tutorial verwendet die Java-Version 1.5.

Die Entwicklung von Java-Anwendungen erfordert theoretisch keine spezielle Entwicklungssoftware und ist mit jedem Texteditor möglich. Dennoch kann diese durch die Verwendung von integrierten Entwicklungsumgebungen (IDEs) erheblich vereinfacht werden. Jeder Programmierer hat seine eigenen Vorlieben, geht es um die Wahl der verwendeten IDE. Dieses Tutorial wurde auf Basis von Eclipse erstellt. Eclipse ist auf der Webseite http://www.eclipse.org/ für alle gängigen Betriebssysteme erhältlich. Natürlich können die hier vorgestellten Tutorials auch mit anderen IDEs wie z.B. Netbeans bearbeitet werden.

1.2 FTP-Client

Der Transfer der erstellten Programme von Ihrem Computer zum Mainframe erfolgt über das FTP-Protokoll. Hierfür benötigen Sie einen entsprechenden FTP-Client. Filezilla ist ein Beispiel für einen kostenlosen FTP-Client, der für alle gängigen Betriebssysteme unter http://www.filezilla.de/ erhältlich ist. Natürlich können Sie auch jeden anderen FTP-Client verwenden. Voraussetzung ist nur, dass dieser SFTP unterstützt.

1.3 TN3270-Emulator

IBM Mainframes verwenden für die Kommunikation 3270-Terminals, welche eine spezielle Variante des Telnet-Protokolls benutzen. Damit Sie sich von Ihrem Computer auf dem Mainframe einloggen können, müssen Sie ein solches Terminal emulieren. Diese Aufgabe übernimmt ein entsprechender TN3270-Emulator. Für Windows existiert das kostenlose Programm wc3270, welches auf der Seite http://x3270.bgp.nu/ zu finden ist. Für Linux kann der x3270-Emulator verwendet werden, der auf selben Seite veröffentlicht ist. Nutzer von MacOs können auf das Programm tn3270 zurückgreifen. Dieses ist auf der Seite http://www.brown.edu/cis/tn3270/index.html der Brown Universität erhältlich.

1.2 Einrichten der Entwicklungsumgebung

Im Folgenden gehen wir davon aus, dass Sie bereits alle oben aufgeführten Hilfsmittel installiert haben und mit ihrer grundlegenden Bedienung vertraut sind. Dieser Abschnitt beschränkt sich daher drauf, Ihnen alle notwendigen weiterführenden Programm- und Projekteinstellungen zu erklären, die für eine funktionsfähige Entwicklungsumgebung benötigt werden. IBM stellt für die Entwicklung von Java-Anwendungen unter CICS eine eigene Bibliothek bereit. Diese enthält die gesamte JCICS-API und ist Teil der Java-Installation des Mainframes. Damit Sie auf Ihrem eigenen System Java-Anwendungen für CICS entwickeln und kompilieren können, müssen Sie zuvor die entsprechende Bibliothek vom Mainframe kopieren und später in alle Java-Projekte einbinden.

Bevor wir die benötige Java-Bibliothek vom Mainframe laden können, müssen wir ein FTP-Client so einrichten, dass wir eine Verbindung zum Mainframe aufbauen können. Auf diesem Wege werden später auch die kompilierten Java-Klassen auf den Mainframe übertragen.

Einrichtung des FTP-Clients und Download der JCICS-Bibliothek

Wir erstellen hierfür eine neue FTP-Verbindung. Das Vorgehen hierfür ist abhängig von dem verwendeten FTP-Programm und daher im entsprechenden Handbuch nachzulesen. Die neue Verbindung sollte folgende Eigenschaften haben:

Protokoll SFTP
Port 22
Server-Adresse 134.2.205.54
Benutzername [Ihr Benutzername]

Im Anschluss können Sie die neu erzeugte Verbindung aufbauen. Diese öffnet Ihr Homeverzeichnis auf dem Mainframe (“/u/prak500”). Die benötigte Bibliothek finden Sie im Verzeichnis “/usr/lpp/cicsts/cicsts31/lib”. Speichern Sie aus diesem Ordner die Datei “dfjcics.jar” auf Ihrem System ab.

Einrichten der Eclipse-Umgebung.

Öffnen Sie zu Beginn eine Workspace Ihrer Wahl. An dieser Stelle empfehlen wir Ihnen, der Übersicht halber eine neue Workspace speziell für diese Reihe von Tutorials anzulegen und in diese Workspace für jedes Tutorial ein eigenes Projekt zu erzeugen. Beim Erzeugen des neuen Projekts, müssen Sie folgende Projekteinstellungen beachten:

- **Als JRE muss die Java JVM 1.5 ausgewählt werden.**
- **Als externe Bibliothek (“External JAR”) muss die zuvor gespeicherte Datei “dfjcics.jar” eingebunden werden.**

Nachdem Sie das neue Java-Projekt erstellt haben, müssen Sie noch sicherstellen, dass die Funktion “Build Automatically” in der Menüleiste unter “Project” aktiviert ist. Diese Einstellung bewirkt, dass nach jeder gespeicherten Änderung automatisch die entsprechende class-Datei erzeugt wird, die wir später auf dem Mainframe laden.

Einrichten der TN3270-Verbindung

Richten Sie innerhalb Ihres TN3270-Emulators eine neue Verbindung mit untenstehenden Eigenschaften ein. Diesen Schritt sollten Sie auf Grund Ihrer Erfahrungen aus anderen CICS-Tutorials bereits selbstständig bewerkstelligen können.

Host-Adresse 134.2.205.54
Portnummer 23
Security SSL
Connection-Type TCP/IP

1.3. Ein- und Ausloggen unter CICS

Die hier vorgestellten Tutorials verwenden CICIS 3.1. Im Folgenden wird beschrieben, wie Sie sich auf dem in Abschnitt 1.6.2 beschriebenen Mainframe unter CICS 3.1 einloggen und ausloggen können.

Starten Sie zu Beginn Ihren TN3270-Emulator und öffnen die in Abschnitt 1.2 beschriebene Verbindung. Sobald die Verbindung erfolgreich aufgebaut wurde, sollte Ihr Emulator den Begrüßungsbildschirm des Mainframes anzeigen. Dieser ist in Abbildung 1 dargestellt.

Einloggen in CICS 3.1 können Sie sich mit der Eingabe:

L CICS1

Nach der Bestätigung der Eingabe erscheint der Begrüßungsbildschirm von CICS, der in Abbildung 1 dargestellt ist. Diesen können Sie mit dem Clear-Befehl (beim wc3270 ist dies Alt+C) Ihres Emulators löschen, um Ihre eigentliche Arbeit unter CICS zu beginnen. Wie Ihnen aufgefallen sein sollte, mussten Sie während des Anmeldeprozesses kein Passwort eingeben. Das CICS 3.1-Subsystem des Mainframes der Universität Tübingen ist so konfiguriert, dass nur für die Verwendung bestimmter Befehle eine Anmeldung mit Benutzername und Passwort notwendig ist. Diese erfolgt über den Befehl:

CESN

Der Login-Befehl wird wie jede andere CICS-Anweisung auch, in die obere, linke Ecke eingetragen. Nach der Eingabe erscheint der Anmeldebildschirm mit der Eingabemaske für Ihren Benutzernamen und Ihr Passwort. Nach der erfolgreichen Anmeldung erscheint ein Bildschirm mit der Nachricht “Sign-on is complete” in der unteren linken Ecke.

Wenn Sie sich in CICS eingeloggt haben, ist es besonders wichtig, dass Sie sich vor Beendigung der Verbindung wieder ordnungsgemäß von CICS abmelden. Sollten Sie dies nicht tun, wird Ihr Account für mehrere Stunden automatisch gesperrt. Abmelden können Sie sich mit dem Befehl:

CESF LOGOFF

Dieser Befehl schließt auch automatisch Ihre Verbindung und Ihr Emulatorfenster.

Abbildung 1: CICS Begrüßungsbildschirm

2. Einführung in JCICS

In diesem Abschnitt werden Sie Ihr erstes JCICS-Programm entwickeln und auf dem Mainframe installieren. Es dient als Grundlage für alle weiterführenden Kapitel, in denen Ihnen verschiedene Techniken der Softwareintegration mit Java und XML unter CICS gezeigt werden. Am Ende dieses Kapitels werden Sie in der Lage sein, unter Anleitung JCICS-Programme auf Ihrem System zu entwickeln und diese anschließend selbstständig auf einem Mainframe unter CICS zu installieren.

2.1. Hello JCICS

In diesem Abschnitt werden Sie eine JCICS-Variante des klassischen HelloWorld-Programms erstellen. Das hier erstellte Programm dient im Wesentlichen dazu Ihnen zu zeigen, wie JCICS-Anwendungen auf dem Mainframe portiert und installiert werden.

Erstellen Sie zu Beginn in Ihrer Entwicklungsumgebung ein neues Java-Projekt “Tutorial1”. Im Anschluss legen Sie innerhalb des Projekts eine neue Java-Klasse “HelloWorld” an. Als Paketnamen sollten Sie Ihren Benutzernamen, gefolgt von “tutorial1”, verwenden. In diesem Beispiel entspricht dieser “prak500.tutorial1”.

Das Listing 1 zeigt Ihnen den gesamten Code der HelloWorld-Klasse. In den folgenden Absätzen werden wir diesen detailliert besprechen und die Unterschiede zu einem HelloWorld-Programm hervorheben, das Sie auf Ihrem eigenen System auf einer Standard-JVM ausführen würden.

Direkt zu Beginn des Programms fällt Ihnen unter (1) auf, dass das Argument der main-Methode nicht wie gewohnt eine Stringarray (String[]) ist, sondern ein Objekt vom Typ CommAreaHolder. Die Commarea, welche durch dieses Objekt gekapselt wird, ist ein speziell reservierter Speicherbereich. Dieser kann von der CICS-Laufzeitumgebung unter anderem für die Ein- und Ausgabe verwendet werden.

In Schritt (2) erzeugen und speichern wir eine Referenz auf das Task-Singelton. Dieses Objekt dient auch der Kapselung der IBM JCICS API und ermöglicht es auf Informationen der Laufzeitumgebung zuzugreifen. Die Implementierung als Singelton ist von der JCICS API vorgegeben und hat den Hintergrund, dass jedes Programm nur über eine einzige Laufzeit verfügt.

Unter CICS ist der Stream System.out nicht mit der Konsole oder einem Terminal verbunden, wie es bei einem heimischen System der Fall wäre. Die Ausgabe in das CICS-Terminalfenster wird über eine Instanz des PrintWriter-Objekts ermöglicht. Diese stellt einen Ersatz für den System.out-Stream dar und kann orthogonal zu diesem verwendet werden. Die PrintWriter-Instanz wird über eine Factory-Methode des Task-Objekts in Schritt (3) bereitgestellt.

Ein weiteres Beispiel für die vom Task-Objekt bereitgestellten Informationen sehen Sie unter Punkt (4). Dort lesen wir aus dem Task-Objekt die Namen des Benutzers und des aktuell ausgeführten Programms mit der dazugehörenden Transaktion aus. Die korrekte Ausführung der in Schritt (4) getätigten Abfrage kann jedoch nicht immer garantiert werden. Aus diesem Grund ist es notwendig, diese Statements in einem try-catch-Block einzufassen, wie Sie ihn bei Punkt (5) sehen können. Dieser soll eventuell auftretende Exceptions abfangen.

2.2. Portierung der Java-Klassen auf den Mainframe

In diesem Abschnitt werden Sie lernen, wie Sie die geschriebenen Java-Programme auf dem Mainframe portieren können. Auf Ihrem System finden Sie nach der Kompilierung für jede Java-Klasse zwei verschiedene Dateien. Eine Datei hat die Dateierweiterung .java und enthält den Quelltext der Klasse. Die zweite Datei trägt die Endung .class und enthält den kompilierten Bytecode, der später von der JVM ausgeführt wird. In welchem Ordner die class-Dateien gespeichert werden, hängt von der verwendeten Entwicklungsumgebung ab. Eclipse und Netbeans zum Beispiel erstellen hierfür einen separaten Ausgabeordner innerhalb der angegebenen Workspace.

Öffnen Sie zunächst den Ordner der verwendeten Workspace in einer Dateiverwaltung Ihrer Wahl (Explorer, Finder...). Dieser enthält für jedes angelegte Projekt einen Unterordner entsprechend des Projektnamens. Eclipse legt innerhalb des Projektordners zwei weitere Ordner an. Einer heißt ”src“ und einer ”bin“. Der bin-Ordner enthält die kompilierten Bytecodevarianten Ihrer Java-Klassen.

Für die spätere Installation der JCICS-Anwendungen müssen Sie diese auf den Mainframe laden. Die kompilierten Klassendateien lassen sich am einfachsten mit Hilfe eines FTP-Klients auf den Mainframe übertragen. Verbinden Sie sich daher über einem FTP-Klienten Ihrer Wahl mit dem Mainframe und navigieren Sie in den Ordner /u/jcicsadm/classes.

Der Ordner /u/jcicsadm/classes wurde als CLASSPATH in das JVM-Profil des Mainframes eingetragen. Sie verfügen mit Ihrem Account nicht über die erforderlichen Rechte, um Ihren eigenen Homeordner als CLASSPATH einzutragen. In den Ordner /u/jcicsadm/classes können Sie nun die gesamte Struktur des bin-Ordners laden. Diese sollte aus zwei Ordnern und einer class-Datei bestehen. Die Ordner wurden beim Kompilieren durch den Java-Compiler automatisch angelegt. Ihre Namen entsprechen den gewählten Paketnamen.

In diesem Beispiel wurde als Paketname “prak500.tutorial1” gewählt. Für diesen Paketpfad wird zuerst ein Ordner “prak500” angelegt, der selbst wiederum einen Ordner “tutorial1” enthält. In diesem befindet sich dann die class-Datei des HelloWorld-Beispiels.

Sollte Ihre Entwicklungsumgebung diese Ordnerstruktur nicht automatisch angelegt haben, ist es notwendig dies auf dem Mainframe per Hand zu tun. Auf jeden Fall sollte die von Ihnen hochgeladene class-Datei am Ende in dem Ordner /u/jcicsadm/classes/prak500/tutorial1 liegen, wobei Sie das “prak500” durch Ihren Benutzernamen ersetzen müssen.

Diese Struktur erlaubt eine saubere Trennung aller Bearbeitungen dieses Kurses. Ihre penible Einhaltung ist aus diesem Grund zwingend erforderlich.

Listing1: HelloWorld in JCICS

```
1  public class HelloWorld
2  {
3    //## (1)
4    public static void main ( CommAreaHolder cah )
5    {
6      //## (2)
7      Task task = Task.getTask();
8      //## (3)
9      PrintWriter out = task.out;
10
11     try {
12       //## (4)
13       out.println("Hello␣" + task.getUSERID() + ",␣welcome␣to␣CICS!");
14       out.println();
15       out.println("This␣is␣program␣" + task.getProgramName());
16       out.println("Transaction␣name␣is␣" + task.getTransactionName());
17     }
18     //## (5)
19     catch (InvalidRequestException e) {
20       e.printStackTrace();
21     }
22   }
23 }
```

2.3. CICS Ressource Definition

CICS verarbeitet Anwendungen mit Hilfe verschiedener Ressource-Definitionen. Für die Installation eines JCICS-Programms als Transaktionen benötigen Sie die Ressourcen:

PROGRAM
TRANSACTION

Die Ressource PROGRAM beschreibt ein unter CICS ausführbares Programm und verweist dafür auf die von Ihnen erstellte Java-Klasse. Eine Transaktion wird über die Ressource TRANSACTION erstellt. Sie kann direkt vom CICS-Transaktionsserver ausgeführt werden und arbeitet selbst wiederum mit der PROGRAM-Definition zusammen.

In diesem Abschnitt lernen Sie, wie Sie beide Ressourcen erstellen, installieren und CICS-Transaktionen ausführen. Für die Bearbeitung der folgenden Schritte ist es notwendig, dass Sie sich unter CICS eingeloggt haben.

52.3.1. Das CICS-Programm

Für die Definition, Bearbeitung, Entfernung und Installation der CICS-Ressourcen ist die eingebaute CICS-Transaktion CEDA zuständig. Für die Definition eines neuen Programms hat der Befehl folgende Syntax:

CEDA DEFINE PROGRAM(Programmname) GROUP(Gruppenname)

Den Programmnamen können Sie frei wählen. Seine maximale Länge beträgt acht Zeichen. Als Gruppennamen sollten Sie Ihren Benutzernamen angeben. In diesem Beispiel wäre dies prak500. CICS unterscheidet bei der Eingabe über das Terminal standardmäßig nicht zwischen Groß- und Kleinschreibung. Die gesamte Eingabe wird stattdessen in Großbuchstaben konvertiert. Für die Definition des HelloWorld-Programms sieht der Befehl wie folgt aus:

CEDA DEFINE PROGRAM(JHELLOW) GROUP(PRAK500)

Nachdem Sie den Befehl durch das Drücken der Entertaste bestätigt haben, öffnet sich eine Liste mit Einstellungsmöglichkeiten der Programmdefinition. Einen Ausschnitt dieser Liste sehen Sie in
Abbildung 1.Wahrscheinlich wird die Liste den Anzeigebereich Ihres Terminals überschreiten. Zum Scrollen können Sie die Tasten F7 und F8 verwenden.

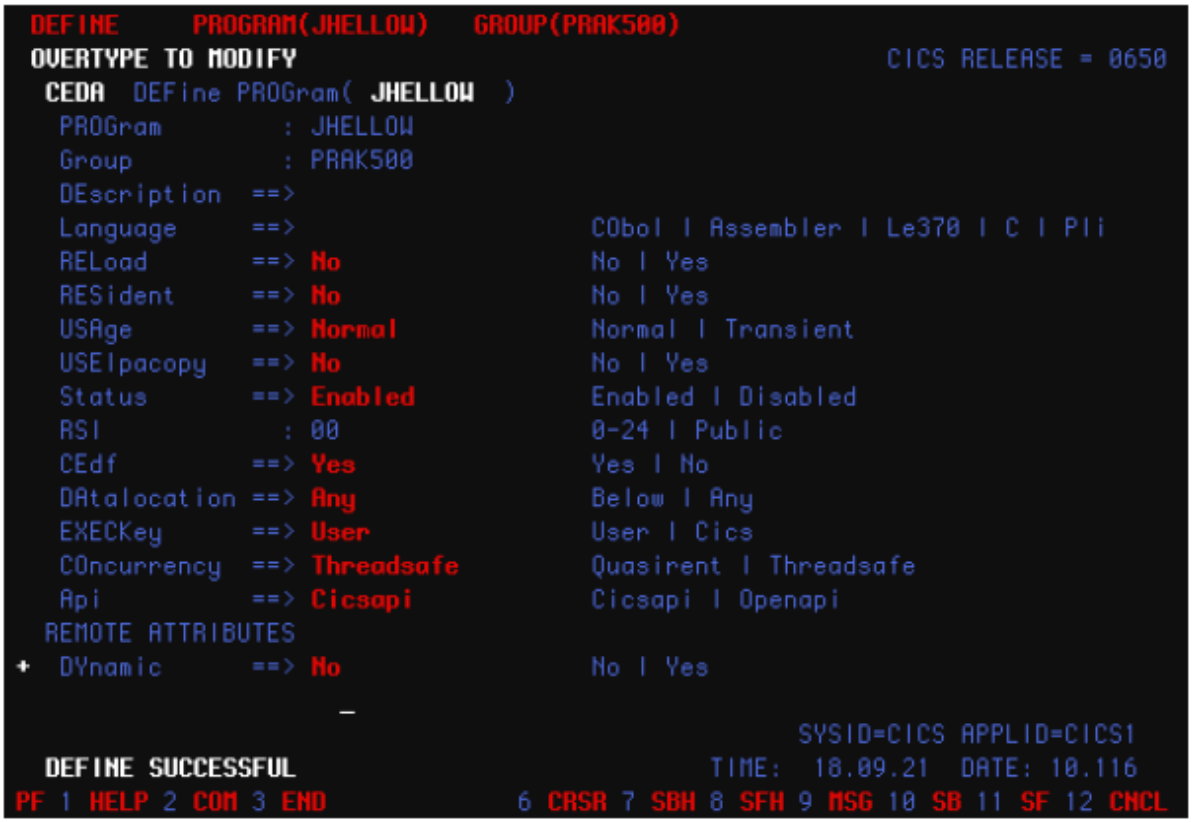

Abbildung 2: Einstellungsmöglichkeiten bei der Definition eines CICS-Programms

Die meisten Einstellungen der Definition müssen nicht geändert werden. Folgende Angaben sind jedoch abweichend von ihren Standardwerten (in Klammern) zu definieren:

Datalocation:	**Any (Below)**
Concurrency:	**Threadsafe (Quasirentrant)**
JVM:	**Yes (No)**
JVMClass:	**Pfad zur Klasse**

In das Feld JVMCLASS müssen Sie den Pfad zu der Java-Klasse angeben, die Sie für Ihre CICS-Anwendung verwenden wollen. In Projekten, die aus mehreren Klassen bestehen, reicht hier die Angabe der Klasse, welche die main-Methode enthält.

Das Feld selbst wird aktiviert, nachdem Sie die Einstellung JVM = Yes gesetzt haben. Im Unterschied zu den anderen Einstellungen wird im Feld JVMCLASS zwischen Groß- und Kleinschreibung unterschieden. Ausgangspunkt des anzugebenden Pfades ist immer der Ordner /u/jcicsadm/classes/. Dies bedeutet, als Pfad zu der HelloWorld-Klasse können Sie Benutzername.tutorial1.HelloWorld eintragen. In diesem Beispiel entspricht dies der Angabe prak500.tutorial1.HelloWorld. Die Angabe der Dateiendung .class ist nicht notwendig. Nachdem Sie alle Einstellungen getätigt haben, können Sie die Definition durch die Betätigung der Eingabetaste speichern. Im Anschluss sollte die Meldung DEFINE SUCCSESSFUL in der unteren, linken Ecke des Terminals erscheinen.

Beenden Sie die CEDA-Umgebung durch Drücken der F3-Taste. Diese agiert ähnlich zu der ESC-Taste unter Windows. Sie erlaubt es Ihnen immer zu der nächst höheren Anwendungsebene zu springen. Auf der höchsten Ebene sehen Sie nur noch die Meldung SESSION ENDED.

Nach der Definition müssen Sie das Programm noch installieren. Dazu geben Sie folgenden Befehl ein:

CEDA INSTALL PROGRAM(JHELLOW) GROUP(Gruppenname)

Im Anschluss sollte die Meldung INSTALL SUCCESSFUL erscheinen. Die Angabe Gruppenname müssen Sie natürlich wieder durch Ihren Benutzernamen ersetzen. In künftigen Installationen müssen Sie JHELLOW durch den Namen des zu installierenden Programms ersetzen.

2.3.2. Die CICS-Transaktion

Theoretisch ist es schon möglich, das oben installierte Programm manuell zu starten. Unser Ziel ist es jedoch, dieses in einer Transaktion einzubinden. Für die Definition einer Transaktion verwenden wir den CEDA-Befehl mit folgender Syntax:

CEDA DEFINE TRANSACTION(Transaktionsname) GROUP(Gruppenname)

Der Name einer Transaktion besteht genau aus vier Zeichen. Auf Grund Ihrer Benutzerrechte ist es Ihnen nur erlaubt Transaktionen auszuführen, die mit einem X beginnen. Wir wählen daher den Namen XJHW. Der Befehl zur Definition sieht daher wie folgt aus:

CEDA DEFINE TRANSACTION(XJHW) GROUP(PRAK500)

Wie bei der Definition eines Programms auch erscheint nach dem Absenden der Anweisung durch Drücken der Eingabetaste eine Liste mit Einstellungsmöglichkeiten. In dieser sind folgende Werte abweichend von ihrer Standardbelegung zu definieren:

Program: Programmname
Taskdataloc: Any

Der hier angegebene Programmname muss mit dem aus Abschnitt 2.3.1 übereinstimmen. Die Definition kann durch die Betätigung der Eingabetaste gespeichert werden. Es sollte wieder die Meldung DEFINITION SUCCESFULL in der unteren linken Ecke des Fensters erscheinen.

Verlassen Sie die Anwendungsebene durch Drücken der F3-Taste. Nach der Definition folgt wieder die Installation. Diese erfolgt durch:

CEDA INSTALL TRANSACTION (XJHW) GROUP(Gruppenname)

Den Gruppennamen müssen Sie wieder durch ihren Benutzernamen ersetzen. In späteren Installationen muss die Angabe XJHW durch den Namen der zu installierenden Transaktion ersetzt werden.

Zum Starten der Transaktion verlassen Sie ggf. alle zurzeit verwendeten Transaktionen. Löschen Sie den aktuellen Bildschirm durch den entsprechenden Befehl Ihres TN3270-Emulators. Geben Sie dann den Namen der Transaktion ein und drücken Sie die Eingabetaste. In diesem Beispiel lautet der Befehl XJHW. Die entsprechende Ausgabe ist in Abbildung 3 dargestellt.

Abbildung 3: Ausgabe der Transaktion

2.4 Zusammenfassung

JCICS-Programme bedienen sich einer anderen Systemschnittstelle als herkömmliche Java-Programme. Zum Beispiel kann die Eingabe einer main-Methode neben einer String-Array auch einen Commareaholder entgegennehmen. Die Ausgabe auf das Terminalfenster erfolgt nicht über den gewöhnlichen System.out-Befehl, sondern über einen PrintWriter. Dieser wird vom Task-Objekt zur Verfügung gestellt, das zusätzlich auch weitere Informationen der Laufzeitumgebung bereitstellt. Nach der lokalen Entwicklung eines Java-Programms müssen die Java-Class-Dateien auf den Mainframe geladen werden. Dies erfolgt am einfachsten mit Hilfe eines FTP-Klienten. Legen Sie die class-Dateien in die entsprechende Unterorder des Java-Classpath.

Für die Ausführung eines Java-Programms unter CICS ist es notwendig, ein CICS-Programm und eine CICS-Transaktion zu erstellen. Beides sind CICS-Ressourcen. Diese müssen zuerst definiert und im Anschluss installiert werden.

Aufgaben

1. Entwickeln Sie ein eigenes Hello-CICS-Programm auf Basis des in vorgestellten Codes. Erweitern Sie die Ausgabe um die Namen Ihrer Teammitglieder.

2. Informieren Sie sich im CICS 3.2 Informationscenter (http://publib.boulder.ibm.com/infocenter/cicsts/v2r3/index.jsp) über weitere Umgebungsvariablen die durch das TASK-Objekt bereitgestellt werden. Nennen Sie mindestens eine weitere und erklären Sie deren Bedeutung.

3. Informieren Sie sich im CICS Transaction Handbuch über weitere Klassen, die der Kapselung der Systemschnittstelle dienen. Nennen Sie eine und erörtern kurz deren Aufgabe.

Die Datei jcicsres01.zip enthält alle Java Resources als zip Archiv. Sie kann heruntergeladen werden unter

www.cedix.de/Vorles/Band3/Resources/jcicsres01.zip