

Tutorial 7

Erstellen eines Servlets unter Unix System Services (USS)

Copyright © Institut für Informatik, Universität Leipzig

ph v/2010/06

Ziel dieses Tutorials ist es, ein Java-Servlet zu erstellen.

Aufgabe: Arbeiten Sie dieses Tutorial durch und lösen Sie exakt alle Aufgaben.

Einige der z/OS-Subsysteme verfügen über eine Benutzer-Shell, in die man sich mit einem geeigneten Klienten einloggen kann. Beispiele sind die TSO-, ISPF- und CICS-Subsysteme.

Ein weiteres Beispiel sind die Unix System Services (USS). Mit Hilfe von USS implementiert z/OS ein dem POSIX-Standard entsprechendes Unix-System mit einem standardkonformen, allerdings auch begrenzten, Funktionsumfang.

USS dient dazu, größere Anwendungen, die unter Unix laufen, auf z/OS zu portieren. Ein Beispiel ist SAP System R/3. Während SAP R/2 ursprünglich für z/OS entwickelt wurde (und den CICS-Transaktionsmonitor benutzt), wurde SAP R/3 ursprünglich für die Unix-Plattformen entwickelt. Durch den USS-Port ist SAP R/3 auch unter z/OS verfügbar.

Ein weiteres Beispiel ist der IBM WebSphere Web Application-Server. WebSphere enthält neben einem Web-Server, eine Servlet Engine, die die Ausführung von Java-Servlets ermöglicht.

Grundsätzlich gibt es für den Einsatz von WebSphere 2 Alternativen:

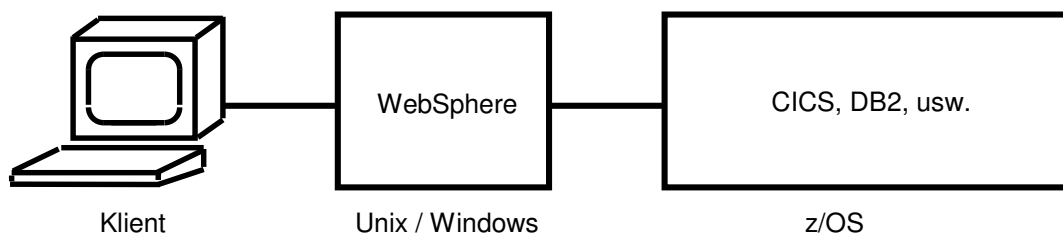


Abbildung 1: 3-Tier-Konfiguration

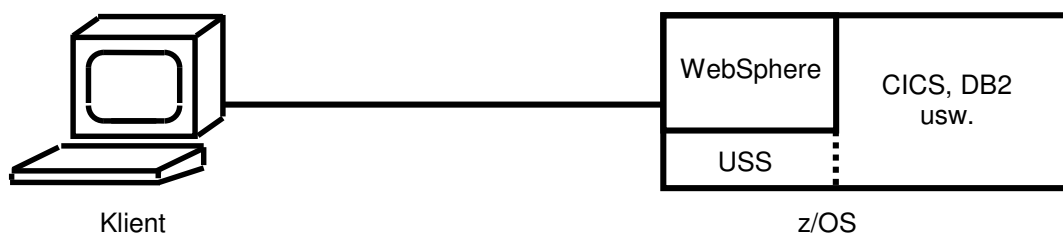


Abbildung 2: 2-Tier-Konfiguration

Abbildung 1 gibt eine 3-Tier-Konfiguration wieder. WebSphere läuft im mittleren Tier auf einem Unix- oder Windows-Rechner.

Abbildung 2 gibt eine logische 3-Tier-, physikalische 2-Tier-Konfiguration wieder. Hier läuft WebSphere als Subsystem auf dem z/OS-Rechner. Bei der "Total Cost of Ownership" dominieren in der Regel die Administrationskosten. In vielen Fällen ist die in Abb. 2 gezeigte Konfiguration hier kostengünstiger.

Wir erstellen ein Servlet, welches unter den z/OS-Unix System Services (USS) läuft.

Um sich ein (sehr einfaches) lauffähiges WebSphere-Servlet zu betrachten, kann einfach folgende Seite besucht werden:

<http://139.18.4.34:9180/prak500/Prak500Servlet>

USS verfügt über zwei unterschiedliche Shells:

- tcsh-Shell, eine Implementierung der Berkley UNIX C csh-Shell.
- z/OS-Shell, ähnlich der UNIX System V-Shell mit einigen Eigenschaften der Korn-Shell.

Die z/OS-Shell ist die Default-Shell.

Ein Zugriff auf beide Shells ist mit Hilfe der rlogin- und telnet-Kommandos möglich. Weiterhin ist es möglich, sich unter TSO einzuloggen, und die z/OS-Shell mit dem TSO-"OMVS"-Kommando aufzurufen. Wir benutzen die letzte Alternative.

```

Menu  Utilities  Compilers  Options  Status  Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                    ISPF Primary Option Menu

0  Settings           Terminal and user parameters           User ID . . : PRAKT20
1  View               Display source data or listings       Time. . . . : 08:37
2  Edit               Create or change source data         Terminal. . : 3278
3  Utilities          Perform utility functions            Screen. . . : 1
4  Foreground         Interactive language processing       Language. . : ENGLISH
5  Batch              Submit job for language processing    Appl ID . . : PDF
6  Command            Enter TSO or Workstation commands     TSO logon  : IKJACCNT
7  Dialog Test        Perform dialog testing                TSO prefix: PRAKT20
8  LM Facility        Library administrator functions      System ID  : DAVI
9  IBM Products       IBM program development products     MVS acct.  : ACCT#
10 SCLM                SW Configuration Library Manager     Release . . : ISPF 4.5
11 Workplace          ISPF Object/Action Workplace

Enter X to Terminate using log/list defaults

Option ==> 6
F1=Help      F3=Exit      F10=Actions  F12=Cancel

```

Abbildung 3 – ISPF Primary Option Menu

Wir loggen uns unter TSO ein und arbeiten uns zum ISPF Primary Option Menu vor. Hier geben wir eine "6" ein, um zum ISPF Command Shell-Panel zu gelangen.

Wir drücken hier Enter und gelangen zum folgenden Bildschirm:

```
Menu List Mode Functions Utilities Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                                     ISPF Command Shell
Enter TSO or Workstation commands below:

===>

Place cursor on choice and press enter to Retrieve command

=> omvs

F1=Help      F3=Exit      F10=Actions  F12=Cancel
```

Abbildung 4 – ISPF Command Shell

Dies ist der ISPF Command Shell Screen. Auf der Kommandozeile können TSO-Kommandos (im Gegensatz zu des ISPF-Kommandos des ISPF-Subsystems) eingegeben werden.

Auf dem ISPF Command Shell Screen sind außerdem die zuletzt eingegebenen Kommandos zu sehen. Hier ist es nur ein einziges Kommando "OMVS". Das "Unix System Services" (USS)-Subsystem hieß früher "Open MVS" (OMVS). Mit dem OMVS-Kommando wechseln wir von dem TSO-Subsystem in das OMVS- bzw. USS-Subsystem.

Wir bewegen den Cursor auf omvs und betätigen die Eingabetaste. Jetzt erscheint "omvs" auf der Kommandozeile. Durch nochmaliges Betätigen der Eingabetaste übergeben wir das OMVS-Kommando an den TSO Command Interpreter. Falls das OMVS-Kommando noch nicht in der Kommandozeile vorhanden sein sollte, ist es einfach oben einzugeben und mit Enter zu bestätigen. Beim nächsten Aufruf der ISPF Command Shell ist es dann wie oben vorhanden. (Abb. 4)

Hierdurch wird die z/OS-Shell der Unix System Services aufgerufen.

```

IBM
Licensed Material - Property of IBM
5647-A01 (C) Copyright IBM Corp. 1993, 1999
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

PRAKT20 : /u/prakt20 >

====>
INPUT
ESC=¢  1=Help      2=SubCmd    3=HlpRetrn  4=Top        5=Bottom    6=TSO
        7=BackScr   8=Scroll   9=NextSess 10=Refresh  11=FwdRetr  12=Retrieve

```

Abbildung 5 – z/OS-Shell

Die Abbildung 5 zeigt die Kommandozeile der z/OS-Shell. Wir befinden uns im Verzeichnis /u/prakt20 des Users PRAKT20.

```

IBM
Licensed Material - Property of IBM
5647-A01 (C) Copyright IBM Corp. 1993, 1999
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

PRAKT20 : /u/prakt20 >

====> mkdir java
INPUT
ESC=¢  1=Help      2=SubCmd    3=HlpRetrn  4=Top        5=Bottom    6=TSO
        7=BackScr   8=Scroll   9=NextSess 10=Refresh  11=FwdRetr  12=Retrieve

```

Abbildung 6 – Erstellen des Verzeichnisses "java"

Die z/OS-Shell ist ähnlich der UNIX System V-Shell mit einigen Eigenschaften der Korn-Shell. Das z/OS Unix System Services "Hierarchical File System" ist ein reguläres Unix-File-System, welches auf z/OS-Datasets abgebildet wird. Hier können wir nun wie in dem uns (hoffentlich) bekannten UNIX-System arbeiten. Zum Erstellen des Java-Servlets legen wir nun einen Ordner "java" an(mit "mkdir java") und betätigen die Eingabetaste (s. Abb. 6).

```

IBM
Licensed Material - Property of IBM
5647-A01 (C) Copyright IBM Corp. 1993, 1999
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

PRAKT20 : /u/prakt20 >mkdir java
PRAKT20 : /u/prakt20 >

====> cd java

ESC=¢      1=Help      2=SubCmd      3=HlpRetrn   4=Top        5=Bottom     6=TSO
           7=BackScr   8=Scroll     9=NextSess  10=Refresh   11=FwdRetr   12=Retrieve
                                                    INPUT

```

Abbildung 7 –Verzeichniswechsel

Nun wechseln wir in dieses Verzeichnis: "cd java" sowie Eingabetaste.
Wir können uns immer mit dem Befehl "ls" den Inhalt des aktuellen Verzeichnisses anzeigen lassen.

Aufgabe: Legen Sie den Ordner "java" an.

```

IBM
Licensed Material - Property of IBM
5647-A01 (C) Copyright IBM Corp. 1993, 1999
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

PRAKT20 : /u/prakt20 >mkdir java
PRAKT20 : /u/prakt20 >cd java
PRAKT20 : /u/prakt20/java >

====> oedit Prakt20Servlet.java

ESC=¢      1=Help      2=SubCmd      3=HlpRetrn   4=Top        5=Bottom     6=TSO
           7=BackScr   8=Scroll     9=NextSess  10=Refresh   11=FwdRetr   12=Retrieve
                                                    INPUT

```

Abbildung 8 – Erstellen des Servlets

Der Quelltext ist 2 Seiten lang, mit F8 kann man runter-, mit F7 hochscrollen.

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT      Prakt20Servlet.java                      Columns 00001 00072
000017   out.println("<h1><center>");
000018   out.println("Hello java world, servlet check ok");
000019   out.println("</center></h1>");
000020   out.println("</body></html>");
000021   }
000022 }
***** ***** Bottom of Data *****

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

```

Abbildung 11 – 2. Teil des Java-Quellcodes

Mit der F3-Taste gehen wir zurück in das OMVS-Subsystem. Dabei wird die Datei automatisch gespeichert. Nun kann man sich mit dem Befehl ls davon überzeugen, ob die Datei angelegt wurde.

```

(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

PRAKT20 : /u/prakt20/java >ls
Prakt20Servlet.java
PRAKT20 : /u/prakt20/java >

====>
ESC=¢      1=Help      2=SubCmd     3=HlpRetrn   4=Top        5=Bottom     6=TSO
           7=BackScr   8=Scroll     9=NextSess  10=Refresh   11=FwdRetr   12=Retrieve
INPUT

```

Abbildung 12 – Der ls-Befehl

Aufgabe: Erstellen Sie Ihr eigenes Java-Servlet. Dabei muß Ihr Name im Servlet angezeigt werden. Bei der Darstellung dürfen Sie gerne kreativ sein. Das Servlet muß den Namen `Prak<xx>Servlet` oder `Prak<xxx>Servlet` (wobei die `xx` bzw. `xxx` für die Nummer Ihres Prakt-Logins stehen sollen) erhalten.

Hinweis zur Bewertung: Je individueller und umfangreicher Sie dieses Servlet gestalten, desto besser wird diese Aufgabe bewertet.

So, jetzt möchten wir unser Programm mit dem normalen Befehl `javac Prakt20Servlet.java` übersetzen. Das geht noch nicht, weil wir vorher die `classpath` Umgebungsvariablen setzen müssen. Wir machen das, indem wir ein Shell-Script "compile" benutzen, welches dies für uns tut, und außerdem den `javac`-Übersetzer aufruft.

```
PRAKT20 : /u/prakt20/java >

===> oedit compile

ESC=¢   1=Help      2=SubCmd    3=HlpRetrn  4=Top        5=Bottom    6=TSO      INPUT
        7=BackScr   8=Scroll    9=NextSess 10=Refresh  11=FwdRetr  12=Retrieve
```

Abbildung 13 – Erstellen des Scriptes "compile" mittels oedit

Dazu erstellen wir ein Scriptfile "compile" in das Verzeichnis `java` mittels `oedit` (s. Abbildung 14).


```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          compile                               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 #!/bin/sh
000002 #
000003 #
000004 export PATH=/usr/lpp/java/J1.4/bin:$PATH
000005 export CLASSPATH=/usr/lpp/java/JSDK2.0/examples:$CLASSPATH
000006 export CLASSPATH=./usr/lpp/java/JSDK2.0/lib/jsdk.jar:$CLASSPATH
000007 javac $1
***** ***** Bottom of Data *****

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

```

Abbildung 14 – Das "compile"-Script

Die Zeilen 000001 bis 000007 enthalten ein normales Unix-Shell-Script. Da wir es bereits vorfinden, brauchen wir es selbst nicht zu schreiben. Das Script setzt die CLASSPATH-Umgebungsvariablen und ruft anschließend den javac-Compiler auf.

Einige alte Servlets *.java- und *.class-Files befinden sich im Directory „./usr/lpp/java/JSDK2.0/examples“. Unser Script ruft den javac-Compiler auf. Zusätzlich stellt es die erforderlichen CLASSPATH-Verbindungen her.

Mit F3 gehen wir zurück zum OMVS-Subsystem.

```

PRAKT20 : /u/prakt20/java >

====> ./compile Prakt20Servlet.java
                                         RUNNING
ESC=¢  1=Help      2=SubCmd    3=HlpRetrn  4=Top      5=Bottom    6=TSO
        7=BackScr  8=Scroll   9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve

```

Abbildung 15 – kompilieren des Servlets

Wir rufen das "compile"-Skript mit unserer Prakt20Servlet.java-Datei auf, um unser Java-Programm zu kompilieren. (Abbildung 15)

```

PRAKT20 : /u/prakt20/java >compile Prakt20Servlet.java
PRAKT20 : /u/prakt20/java >

====>
                                         RUNNING
ESC=¢  1=Help      2=SubCmd    3=HlpRetrn  4=Top      5=Bottom    6=TSO
        7=BackScr  8=Scroll   9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve

```

Abbildung 16 – Die (ausbleibende) Ausgabe von "compile"

Das Ausbleiben von Fehlermeldungen weist auf eine erfolgreiche Übersetzung hin (s. Abbildung 16).

```
PRAKT20 : /u/prakt20/java >compile Prakt20Servlet.java
PRAKT20 : /u/prakt20/java >ls
Prakt20Servlet.class  Prakt20Servlet.java  compile
PRAKT20 : /u/prakt20/java >
```



```
====>
```

ESC=¢	1=Help	2=SubCmd	3=HlpRetrn	4=Top	5=Bottom	6=TSO	RUNNING
	7=BackScr	8=Scroll	9=NextSess	10=Refresh	11=FwdRetr	12=Retrieve	

Abbildung 17 – Anzeige von "Prakt20Servlet.class"

Ob der Übersetzungsvorgang erfolgreich war, kann man mit dem Befehl ls herausfinden. War die Übersetzung erfolgreich, existiert die Datei Prakt20Servlet.class (s. Abbildung 17).

Aufgabe: Legen Sie das Script "compile" an und übersetzen Sie Ihr java-Servlet damit erfolgreich.

Nun muss diese Datei nur noch mit einem einfachen http-Server, der einen Servlets ausführen kann, gestartet werden, damit sie beim Aufruf über das Internet auf dem z/OS-Mainframe ausgeführt werden kann.

Durch folgende Aufrufe wird ein einfacher java http-Server gestartet, wobei <xxx> unsere als dreistellige Zahl interpretierte Prak(t)<xxx> Nummer ist. Zunächst werden die Umgebungsvariablen PATH und CLASSPATH exportiert, dann wird durch Aufruf von java, gefolgt von Mainklasse und Portadresse des Httpservers der eigentliche Start durchgeführt.

```
export PATH=/usr/lpp/java/J1.4/bin:$PATH
export CLASSPATH=./usr/lpp/java/JSDK2.0/lib/jsdk.jar ((Zeile fortsetzen))
/usr/lpp/java/JSDK2.0/examples:$CLASSPATH
java sun.servlet.http.HttpServer -p 51<xxx> &          („&“ nicht vergessen)
```

Im Terminalfenster sieht das dann etwa wie Abbildung 18 aus:

```
===> export PATH=/usr/lpp/java/J1.4/bin:$PATH
===> export CLASSPATH=./usr/lpp/java/JSDK2.0/lib/jsdk.jar:/usr/lpp/java
      /JSDK2.0/examples:$CLASSPATH

===> java sun.servlet.http.HttpServer -p 51<xxx> &
```

ESC=ç	1=Help	2=SubCmd	3=HlpRetrn	4=Top	5=Bottom	6=TSO
	7=BackScr	8=Scroll	9=NextSess	10=Refresh	11=FwdRetr	12=Retrieve

Abbildung 18 – Starten des Servlets

Wenn dieser Aufruf erfolgreich war, können wir endlich das fertige Ergebnis (unser Servlet) betrachten.

Dies geschieht unter Berücksichtigung unserer jeweiligen Startparameter durch den Aufruf von:

<http://139.18.4.34:51<xxx>/servlet/Prakt20Servlet>

Aufgabe: Führen Sie Ihr Servlet wie oben beschrieben aus.

Aufgabe: Erzeugen Sie einen Screenshot Ihres Browsers, der Ihr Servlet anzeigt. Der Screenshot sollte im kompakten JPG-Format erstellt werden und darf eine Größe von 90 KByte nicht überschreiten. Sollte die auf Ihrem PC installierte Software das JPG-Format nicht unterstützen, ist auch das Bitmap-Format erlaubt (max. Bildgröße 350 KByte).

Aufgabe: Schicken Sie eine Mail an Ihren Tutor mit den folgenden Inhalten:

- 1) Ein Link auf Ihr Servlet, damit Ihr Tutor Ihr Servlet aufrufen könnte, z.B.
<http://139.18.4.34:51<xxx>/servlet/Prak<xxx>Servlet>
- 2) Den erzeugten Screenshot als Anhang.

Anmerkung: Falls Sie Ihr Servlet modifizieren, könnte es sein, dass der Browser nicht die neueste Version Ihres Servlets anzeigt. Löschen Sie dann einfach den Cache-Speicher Ihres Browsers.

Empfehlung: Damit wir nicht bei jedem Einloggen in USS die Pfade wieder neu eintippen müssen, können wir uns auch, wie unter UNIX üblich, eine .profile Datei erstellen. Diese sollte etwa so aussehen:

```
#
#
export PS1='$LOGNAME': '$PWD': ' >'
#
export PATH=/usr/lpp/java/J1.4/bin:$PATH
#
export CLASSPATH=./usr/lpp/java/JSDK2.0/lib/jsdk.jar:/usr/lpp/java
((Fortsetzung der vorigen Zeile)) /JSDK2.0/examples:$CLASSPATH
```

Der Start erfolgt nach dem Anmelden im OMVS mit “. .profile“ und darauf folgend:

```
cd java
java sun.servlet.http.HttpServer -p 51<xxx> &
```

Nachdem wir die Funktion unseres Servlets überprüft haben, beseitigen wir den http-Server Prozess, indem wir zunächst mit:

```
ps -u prak(t)<xxx>
```

die ProzessID dieses Java-Prozesses ermitteln. Wir erhalten etwa:

```
33620034 ttyp0000 0:00 /usr/lpp/java/J1.4/bin/java
```

Mit:

```
kill 33620034
```

beenden wir den Java-Prozess des http-Servers.

Wir verlassen das OMVS-Subsystem, indem wir auf der Kommandozeile "exit", gefolgt von Enter eingeben (s. Abbildung 19).

```
PRAKT20 : /u/prakt20/java >
PRAKT20 : /u/prakt20/java >ls
Prakt20Servlet.class  Prakt20Servlet.java  compile
PRAKT20 : /u/prakt20/java >
PRAKT20 : /u/prakt20/java >

====> exit

ESC=¢   1=Help      2=SubCmd    3=HlpRetrn  4=Top        5=Bottom    6=TSO
        7=BackScr   8=Scroll   9=NextSess 10=Refresh  11=FwdRetr  12=Retrieve

RUNNING
```

Abbildung 19 – Exit

```

PRAKT20 : /u/prakt20/java >
PRAKT20 : /u/prakt20/java >ls
Prakt20Servlet.class  Prakt20Servlet.java  compile
PRAKT20 : /u/prakt20/java >cp Prakt20Servlet.class /web/serv1/AplServer/servlets
PRAKT20 : /u/prakt20/java >exit

>>>> FSUM2331 The session has ended.  Press <Enter> to end OMVS.
====>

                                RUNNING
ESC=¢   1=Help      2=SubCmd    3=HlpRetrn  4=Top      5=Bottom   6=TSO
        7=BackScr  8=Scroll   9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
    
```

Abbildung 20 – Bestätigung, dass die OMVS-Sitzung beendet wurde

Anschließend ist noch einmal Enter zu betätigen, um OMVS zu verlassen.

```

Menu List Mode Functions Utilities Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                                ISPF Command Shell
Enter TSO or Workstation commands below:

====>

Place cursor on choice and press enter to Retrieve command

=> omvs
=> omvs

F1=Help      F3=Exit      F10=Actions  F12=Cancel
    
```

Abbildung 21 – ISPF Command Shell

Es erscheint der ISPF Command Shell Screen wieder. Die Sitzung wird durch dreimal F3, "3" einschließlich Enter und abschließend durch "logoff" einschließlich Enter beendet.