

# COBOL

## Dumpanalyse im z/OS

Musterlösungen zu den Übungen

23. Januar 2009

Eine Ausarbeitung von:

**cps4it**

Ralf Seidler • Stromberger Straße 36A • 55411 Bingen  
Fon: +49-6721-992611 • Fax: +49-6721-992613 • Mail: [ralf.seidler@cps4it.de](mailto:ralf.seidler@cps4it.de)  
Internet : <http://www.cps4it.de>  
Steuernummer: 08/220/2497/3, Finanzamt Bingen, Ust-ID : DE214792185

Diese Seite bleibt frei

## Inhaltsverzeichnis

<b>1</b>	<b>Vorbereitungen</b>	<b>5</b>
1.1	Anmelden und Test der User-Iden	5
1.2	Zugang zu Bookmanager im Intranet	5
1.3	Zugang zum Bookmanager im Internet	5
1.4	Wichtige Books in COBOL	6
1.5	Wichtige Books in LE	6
1.6	Das Bookshelf „Messages and Codes“	6
<b>2</b>	<b>LE – Programm Management</b>	<b>7</b>
2.1	Sprachen unter LE	7
2.2	CEL	7
2.3	Enklave	7
2.4	Thread	7
2.5	Prozess	7
<b>3</b>	<b>LE – Condition Handling</b>	<b>8</b>
3.1	Beispiele der CEL	8
3.2	Initialisierung	8
3.3	Storage Manager	8
3.4	Condition Handler	8
3.5	Terminator	8
<b>4</b>	<b>LE – Condition Handling</b>	<b>9</b>
4.1	Wichtige Module im Dump	9
4.2	Aufbau der COBOL-Meldungen	9
4.3	Dumpinformationen finden in Beispiel	9
4.4	Dumpinformationen finden in einem Produktionsabbruch	11
<b>5</b>	<b>Linkage Convention und Optionen</b>	<b>12</b>
5.1	Programm erstellen	12
5.2	Umwandlungsoptionen einstellen	12
5.3	JCL zu Programm erstellen	12
<b>6</b>	<b>Linkage Convention und Optionen</b>	<b>13</b>
6.1	Steuerblöcke in Umwandlungsliste	13
6.2	Tests des Programms mit verschiedenen Abbrüchen	13
6.2.1	Abbruch SOC7 auf verschiedenen Feldern	13
6.2.2	Abbruch SOCB	13
6.2.3	Abbruch SOC4 mit Subscript	13
6.2.4	Abbruch U1066 mit Subscript	13
6.3	Besonderes im IMS	13
<b>7</b>	<b>Programmiertechniken</b>	<b>14</b>
7.1	Index-Nutzung	14
7.2	Variablendefinitionen	14

Diese Seite bleibt frei

## 1 Vorbereitungen

---

Zum Schmunzeln

**"Glück ist: zu begreifen, wie alles zusammenhängt."**

Sten Nadolny (\*1942), dt. Schriftsteller

### 1.1 Anmelden und Test der User-Iden

---

Keine Musterlösung

### 1.2 Zugang zu Bookmanager im Intranet

---

Eine Möglichkeit ist über das Prodnet zu gehen; dort gibt es verschiedene Möglichkeiten, um zum Bookmanager zu kommen.

### 1.3 Zugang zum Bookmanager im Internet

---

<http://www.ibm.com>

Nach Eingabe des Suchbegriffes „COBOL“ rechts oben erhält man eine Liste vieler Seiten. Die erste Angabe verweist auf:

<http://www-306.ibm.com/software/awdtools/cobol/>

Links ist eine Navigationsleiste mit dem Begriff Library. Nach dem Klick, „Enterprise-COBOL“ auswählen. Die neueste Version ist *Enterprise COBOL for z/OS, Version 3 Release 4*

Geben Sie als Suchbegriff „Language Environment“ ein.

Auf der Liste gibt es einen Link, der direkt zur Library führt. Die aktuellste Version steht oben und ist z/OS V1R7 Language Environment.

Die Einstiegsseite zu LE ist übrigens:

<http://www-03.ibm.com/servers/eserver/zseries/zos/le/>

## 1.4 Wichtige Books in COBOL

---

Die wesentlichen Books sind:

[Enterprise COBOL for z/OS V3.4 Language Reference](#)

[Enterprise COBOL for z/OS V3.4 Programming Guide](#)

## 1.5 Wichtige Books in LE

---

Die wesentlichen Books sind:

[z/OS V1R7.0 Language Environment Concepts Guide](#)

[z/OS V1R7.0 Language Environment Programming Guide](#)

[z/OS V1R7.0 Language Environment Run-Time Messages](#)

Alle anderen Books sind themenspezifisch.

## 1.6 Das Bookshelf „Messages and Codes“

---

Auf der IBM-Seite sind diese nicht einfach zu finden. Die neueste Version liegt bereit unter

<http://publibfp.boulder.ibm.com/cgi-bin/bookmgr/Shelves/ez2mz700>

Der Weg dauert recht lange und ist wie folgt möglich:

Einstieg über: <http://publibfp.boulder.ibm.com/cgi-bin/bookmgr/library>

Als Suchbegriff Messages eingeben. Am Ende der Liste ist das Gesuchte.

## **2 LE – Programm Management**

---

### **2.1 Sprachen unter LE**

---

Language Environment ist die gemeinsame Laufzeitumgebung für COBOL, PL1, C/C++ und Fortran. Assembler-Module können ebenfalls aufgerufen werden, denn ASM benötigt keine Laufzeitumgebung.

### **2.2 CEL**

---

Die CEL ist die Sammlung aller Laufzeitmodule unter LE (Common Execution Library). Sie beinhaltet Funktionen zur Initialisierung, die Terminierung, das Storage Management, Condition Handling, Message Services, Datums- und Zeitfunktionen sowie mathematische Funktionen.

### **2.3 Enklave**

---

Enklave ist innerhalb eines Prozesses die Laufzeitumgebung mit allen dazugehörigen Anwendungsprogrammen.  
Ein Thread mit einem COBOL-Hauptprogramm ist mit seiner Umgebung eine Enklave.

### **2.4 Thread**

---

Ein Thread ist ein Hauptprogramm mit allen seinen Unterprogrammen und seinen (jeweils) eigenen Daten.

### **2.5 Prozess**

---

Ein Prozess ist die Summe aller Enklaven, die einen fachlichen Vorgang bearbeiten. Ein Prozess kann einen weiteren Prozess anstoßen, hat aber keinerlei Verbindungen und Daten miteinander.

## **3 LE – Condition Handling**

---

### **3.1 Beispiele der CEL**

---

Siehe Lösung der Aufgabe 2.2.  
Oder auch: CEEOCT, CEEDATE, CEEMOUT und viele andere mehr.

### **3.2 Initialisierung**

---

Umgebung aufbauen, Speicher besorgen (lassen), Condition Handler initialisieren, Optionen prüfen und sprachspezifische Laufzeitumgebung aufbauen; Übergabe der Kontrolle an das Anwendungshauptprogramm. Unter IMS gilt überigens der IMS-Regioncontroller für LE als das Anwendungshauptprogramm.

### **3.3 Storage Manager**

---

Heap-Storage besorgen, Stack-Storage besorgen; jeweils zu Beginn und dann während der Ausführungszeit explizit und implizit.

### **3.4 Condition Handler**

---

Abfangen der Fehlersituationen; prüfen, was zu tun ist (resume, percolate, promote, fix-up and resume); Informationen sammeln und ausgeben.

### **3.5 Terminator**

---

Vollständiges „garbage collection“ nach Ende der Anwendung.

oder: Gouverneur von Kalifornien? ;-)



## 4 LE – Condition Handling

### 4.1 Wichtige Module im Dump

CEEHDSP (Top-Modul), CEEPLPKA (LE-Hauptprogramm), CEEBINIT (Initialisierungsmodul des LE; taucht nur auf, wenn in der Initialisierungsphase etwas schief geht), CEEHSGLT (signal handler), CEEV#GH (getheap), CEEV#FH (freeheap), CEEV005 (COBOL event handler)

### 4.2 Aufbau der COBOL-Meldungen

IGZnnnnx mit x=I,W,E,S,C

### 4.3 Dumpinformationen finden in Beispiel

- Kurzinformation über den Dump

```
CEE3211S The system detected a decimal-divide exception (System Completion Code=0CB).
        From compile unit TES47 at entry point TES47 at compile unit offset +000004A8 at entry offset +000004A8 at
        address 2F717090.
<> LEAID ENTERED (LEVEL 05/09/2005 AT 11.27)
<> LEAID PROCESSING COMPLETE. RC=0
```

- Call Hierarchie

```
Traceback:
 DSA Addr  Program Unit  PU Addr  PU Offset  Entry      E Addr  E Offset  Statement  Load Mod  Service  Status
0001D458  CEEHDSP      0A997968 +00004904 CEEHDSP    0A997968 +00004904 CEEPLPKA  UK06547  Call
0001D318  TES47        2F716BE8 +000004A8 TES47      2F716BE8 +000004A8 TES47     Exception
0001D130  IGZCFCC      0A8429A8 +000002CA IGZCFCC    0A8429A8 +000002CA IGZCPAC   Call
0001D018  TES39        2F700CF8 +00000468 TES39      2F700CF8 +00000468 TES39     Call
```

- Fehlerart und Meldung

```
Condition Information for Active Routines
Condition Information for TES47 (DSA address 0001D318)
CIB Address: 0001DD98
Current Condition:
CEE3211S The system detected a decimal-divide exception (System Completion Code=0CB).
```

- Inhalt des Programm-Status-Word

```
Machine State:
ILC..... 0006      Interruption Code..... 000B
PSW..... 078D1000 AF717096
```

- Abbruchadresse

```
Location:
Program Unit: TES47 Entry: TES47 Statement: Offset: +000004A8
```



- Informationen über Variableninhalte der Anwendungsprogramme

```
Local Variables:  
TES47 was not compiled with the SYM suboption of the TEST option. A formatted variable dump cannot be produced.
```

```
Local Variables:  
TES39 was not compiled with the SYM suboption of the TEST option. A formatted variable dump cannot be produced.
```

## 4.4 Dumpinformationen finden in einem Produktionsabbruch

---

Keine Musterlösung

## 5 Linkage Convention und Optionen

---

### 5.1 Programm erstellen

---

Falls eine Kodierung zu lange dauert, kopieren Sie die Programme von der Datei des Referenten.

### 5.2 Umwandlungsoptionen einstellen

---

Die Optionen sollten i.A. wie der Default eingestellt sein. Hilfreiche Optionen für den Test sind: LIST, TEST(SYM), SSRANGE.

Die Optionen in der Produktion sollten stets NOTEST und NOSSRANGE sein, da sonst deutliche Performanceeinbußen zu erwarten sind.

LIST:	COBOL-Befehle mit Adresse und Assembler-Auflösung
NOLIST:	COBOL-Befehle nur mit Adresse und Zeile
TEST(SYM):	Bei Abbruch Anzeige der Variableninhalte
NOTEST:	„normaler“ Dump
SSRANGE:	Prüfung auf Index / Subscript-Überlauf / Bereiche
NOSSRANGE:	keine Prüfung auf Indizes / Subscripte / Bereiche

### 5.3 JCL zu Programm erstellen

---

Falls eine Kodierung zu lange dauert, kopieren Sie die JCL von der Datei des Referenten.

## **6 Linkage Convention und Optionen**

---

### **6.1 Steuerblöcke in Umwandlungsliste**

---

Zu finden sein sollten zumindest BLW, BLL, BLX, TGT.

### **6.2 Tests des Programms mit verschiedenen Abbrüchen**

---

Führen Sie das Programm aus mit unterschiedlichen Abbruchkonstellationen. Versuchen Sie, die Abbruchursache in den Dump-Informationen zu finden. Suchen Sie auch die im Kurs erwähnten Steuerblöcke. Was helfen Ihnen diese bei der Suche nach der Abbruchursache? Wann helfen Sie Ihnen?

#### **6.2.1 Abbruch S0C7 auf verschiedenen Feldern**

Keine Musterlösung

#### **6.2.2 Abbruch S0CB**

Keine Musterlösung

#### **6.2.3 Abbruch S0C4 mit Subscript**

Es fällt auf, dass der Abbruch nicht mit einem Subscript auftritt, der „knapp über der Grenze“ liegt, sondern erst bei einem sehr hohen Wert. Dies ist ein Hinweis, dass der eigene Speicher „beliebig“ zerschossen werden darf, ehe das System sich meldet. Und genau in solchen Fällen kann das Wissen um die Steuerblöcke die entscheidenden Hinweise auf die Abbruchursache geben.

#### **6.2.4 Abbruch U1066 mit Subscript**

Keine Musterlösung

### **6.3 Besonderes im IMS**

---

Folgende Punkte:

- Hauptprogramm ist IMS-Region-Controller DFSRCCnn
- Es gibt im DUMP Informationen zu jedem PCB.
- IMS-Fehler sind U-Codes.

## 7 Programmiertechniken

---

### 7.1 Index-Nutzung

---

Sie sollten den Block finden, in dem der Index zu finden ist.

### 7.2 Variablendefinitionen

---

Überlegen Sie sich die Auswirkungen im Speicherinhalt bei der Nutzung der folgenden Definitionen:

- REDEFINES  
gleiche Adresse; ersichtlich an BLW-Informationen
- 88-er Stufen  
gleiche Adresse; ersichtlich an BLW-Informationen
- nicht benutzte Variablen  
Hinweis: BLWxxxx in der Umwandlung
- SYNCHRONIZED  
Es gibt „leere“ Bereiche im Dump. Wichtig dabei ist, die Adressen aus der Umwandlungsliste zur Hand zu haben.