

BACEMT

Installation und Handhabung

Dieses Handbuch wird als Anleitung für die Installation und Handhabung von BACEMT herausgegeben.

Anschrift: Lattwein GmbH
Otto-Brenner-Straße 25
52353 Düren

Telefon: 02421 81051

Telefax: 02421 82127

Internet: <http://www.lattwein.de>

E-Mail: <mailto:service@lattwein.de>

Auskunft: In der unten angegebenen Arbeitszeit steht Ihnen ein zentraler Telefon-Wartungsdienst zur Verfügung, der bemüht ist, alle Fragen zu beantworten, die in diesem Handbuch nicht behandelt wurden.

Arbeitszeit: 8:30 - 16:00 Uhr

Installationsprozeduren

Vor der Bandinstallation müssen folgende Prozeduren angelegt werden:

```
CATALOG CPGINST1.PROC R=Y      EOD=/+
// DLBL Lib,'Kennsatz',,
// EXTENT .....
// EXEC LIBR,PARM='ACCESS SUBLIB=Lib.Sublib'
/+
```

Die Prozedur CPGINST1 wird immer aufgerufen, wenn Module mit dem Typ SOURCE oder OBJ in die Library gespeichert werden. Es muss für diese Funktionen eine gültige ACCESS-Anweisung in der EXEC-LIBR-Karte angelegt werden. Die DLBL- und EXTENT-Anweisung ist nicht erforderlich, soweit sie im Standardlabel oder Partitionlabel angelegt ist.

```
CATALOG CPGINST2.PROC R=Y      EOD=/+
// DLBL Lib,'Kennsatz',,
// EXTENT .....
// LIBDEF SOURCE,SEARCH=(Lib.Sublib)
// LIBDEF OBJ,SEARCH=(Lib.Sublib)
// LIBDEF PHASE,CATALOG=Lib.Sublib,SEARCH=(Lib.Sublib)
/+
```

Die Prozedur CPGINST2 wird immer aufgerufen, wenn Module mit dem Typ Phase in die Library katalogisiert werden. Es muss für diese Funktion eine gültige CATALOG-Anweisung angelegt werden. Die DLBL- und EXTENT-Anweisung ist nicht erforderlich, soweit sie im Standard- oder Partitionlabel angelegt ist.

Bitte tragen Sie für "Lib" die Library und für "Sublib" die Sublibrary ein, in die BACEMT installiert werden soll.

Die OBJ-Search-Anweisung muss folgende Libraries aufweisen:

```
BACEMT Library, IJSYSRS.SYSLIB, PRD1.BASE
```

Installation des Bandes

BACEMT wird dem Anwender in Form eines Magnetbandes bzw. einer Magnetbandkassette zugestellt.

Das Magnetband ist geblockt (3440/80) und enthält alle für die Katalogisierung der Programme erforderlichen Steuerkarten.

Das Band kann direkt gelesen werden. Dazu wird das Band auf eine freie Bändeinheit montiert. Der Operator gibt den Befehl

```
S RDR,Cuu
```

ein, wobei C den Kanal und uu die Einheit, z. B. 280, angibt.

Sollen die Jobs in eine bestimmte Reader-Klasse abgestellt werden, so ist die gewünschte Klasse mit anzugeben:

```
S RDR,Cuu,X
```

Eintragungen in die CICS-Tabellen

Für die Ausführung von BACEMT sind folgende Erweiterungen der CICS-Tabellen erforderlich:

Die CICS PPT muss um das Copy-Buch BACPPT und die CICS PCT muss um das Copy-Buch BACPCT ergänzt werden.

Bitte nehmen Sie die Copys in die jeweilige Tabelle auf und wandeln Sie beide Tabellen um.

Nach dem CICS-Startup kann BACEMT benutzt werden.

Starten von BACEMT

BACEMT muss in jedem CICS installiert werden, in dem es aufgerufen werden soll. BACEMT muss entweder nach jedem CICS-Startup manuell gestartet werden (siehe nächste Seite) oder wird durch einen Eintrag in der PLT immer automatisch aktiviert.

Automatisch aktivieren (Definition in PLT)

Wird die Aktivierungstransaktion in der PLT definiert, so wird BACEMT nach jedem CICS-Startup automatisch gestartet.

Bei der Definition muss folgendes beachtet werden:

1. In der SIT müssen die Parameter

```
PLTPI=XX    und  
PLTSD=YY
```

gesetzt werden.

2. In der PPT müssen die Einträge

```
DFHPPT TYPE=ENTRY,PROGRAM=DFHPLTXX  
DFHPPT TYPE=ENTRY,PROGRAM=DFHPLTY Y
```

aufgenommen werden.

3. In der DFHPLTXX wird folgender Eintrag vorgenommen:

```
DFHPLT TYPE=ENTRY,PROGRAM=BACSTA
```

4. In der DFHPLTY Y wird folgender Eintrag vorgenommen:

```
DFHPLT TYPE=ENTRY,PROGRAM=BACSTA
```

Dieser Eintrag muss vor dem Eintrag

```
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
```

definiert werden.

Starten von BACEMT

Manuell starten

Wenn keine PLT-Einträge vorgenommen wurden, muss BACEMT nach jedem CICS-Startup gestartet werden.

Zum Starten von BACEMT muss auf einem leeren Bildschirm "BACS START" eingegeben werden, lediglich Operator-Commands können ohne den Befehl "BACS START" abgesetzt werden.

Es wird die Meldung

```
MSGM01 BATCH ATTACH SUCCESSFUL
```

am Bildschirm angezeigt.

Ab diesem Zeitpunkt kann das Programm BACEMT in diesem CICS CEMT-Befehle ausführen. Es werden alle CEMT-Befehle, die möglich sind, an das CEMT-Programm DFHEMTA abgesetzt.

Wenn der Befehl erfolgreich verarbeitet wurde, wird der Returncode auf 0 gesetzt. Der Returncode kann im Conditional Job Control abgefragt werden.

File Transfers automatisieren

Mit BACEMT können File Transfers automatisiert werden. Im Job Stream können z. B. POWER-Listen an einen bestimmten Rechner übertragen oder File Transfers gestartet werden.

Der gesamte Befehlsvorrat des QIT steht im BACEMT zur Verfügung.

auf TCP/IP-Basis

```
// EXEC BACEMT
BCEF5 INIT QCFH 00      T000.000.010.000.000.019:01414 -
C:\QIT\QIT.EXE          -
/C=C /F=CPGESD /K=ALLE /O=QPGUK.JOB
/*
```

Aus dem Batchbetrieb wird über die Transaktion QCFH in der Partition F5 am Knoten APPB bzw. an der IP-Adresse 10.0.0.19 am Port 1414 für den Bildschirm T212 ein File Transfer CPGESD gestartet. Der File Transfer arbeitet vom Host zum PC (C=C), überträgt alle Keys (K=ALLE) und stellt das Ergebnis im QIT-Default-Verzeichnis als QPGUK.JOB ab.

Beachte:

- Die Port Number muss fünfstellig angegeben werden.
- Die IP-Adresse muss ebenfalls entsprechend dem Beispiel angegeben werden.
- Die Tabelle der Parameter finden Sie am Ende des QIT-Handbuchs.

CEMT-Befehle

Mit BACEMT können aus der Batch-Partition CEMT-Befehle abgesetzt werden.

Beispiel:

```
// JOB BATCH
// EXEC BACEMT
BCEF5 CEMT SET FILE(DATNAME) CLO
/*
// EXEC UPDATE
/*
      :
      :
// EXEC BACEMT
BCEF5 CEMT SET FILE(DATNAME) OPE
/*
/&
```

Aufbau BACEMT-Befehl

- Stelle 1 - 3 Konstante BCE (WAIT 2 Minuten) oder
Konstante BCL (Long Task 8 Minuten)
- Stelle 4 - 5 CICS-Partition (hier: F5)
- Stelle 7 - 80 CEMT-Befehl, der sonst am Bildschirm
eingegeben würde.
- Stelle 7 - 12 kann auch eine SYSID beinhalten (in \$
Zeichen eingeschlossen, um über LU6.2 in
einem anderen CICS Befehle abzusetzen.)

Beispiel:

```
BCEF5 $CICP$ CEMT SET FILE(DATNAME) CLO
```

Dieser Befehl wird in einem CICS mit der SYSID: CICP durchgeführt, das in der gleichen oder einer anderen CPU definiert ist. Es muss über VTAM eine LU6.2 Verbindung zu diesem CICS bestehen.

Operator-Konsol-Befehle

Mit BACEMT können aus der Batch-Partition Operator-Konsol-Befehle abgesetzt werden.

Beispiel:

```
// JOB START
// EXEC BACEMT
CMD    R RDR,BATCHJOB
/*
/ &
```

Aufbau des BACEMT-Befehls

Stelle 1 - 3 Konstante CMD
Stelle 7 - 80 Operator-Befehl, der sonst an der System-Konsole gegeben würde.

Beachte:

Ein CICS-Shutdown-Befehl kann nur über Konsolbefehle gestartet werden.

Beispiel:

```
// JOB ENDCICS
// EXEC BACEMT
CMD    MSG F2
CMD    2 CEMT PER SHUT IMM
/*
/ &
```

Operator-Konsol-Befehle Fortsetzung

Zur Automatisierung kann eine Batch-Partition eine vorher definierte Zeit warten, bevor der nächste Befehl erfolgt.

Beispiel:

```
// JOB START SQL
// EXEC BACEMT
CMD    R RDR,SQLSTART
BWAIT 0100
CMD    MSG F5
CMD    5 CIRB CICSPROD
BWAIT 0010
CMD    5
/*
/ &
```

Aufbau des BACEMT-Befehls

Stelle 1 - 5 Konstante BWAIT (= BATCH WAIT)
Stelle 7 - 8 Minuten, die gewartet werden sollen
Stelle 9 - 10 Sekunden, die gewartet werden sollen

Der oben aufgeführte Job gibt den Startjob für SQL frei, wartet eine Minute und führt dann den CIRB-Befehl in der CICS-Partition durch, wartet dann noch 10 Sekunden und gibt im F5 die Konsole frei.

CICS-Task

Mit BACEMT kann eine CICS-Task vom Batch aus gestartet werden.

Beispiel:

```
// JOB TASK
// EXEC BACEMT
BCEF7 INIT TRAN Hier können Daten übergeben werden.
/*
/ &
```

Aufbau des BACEMT-Befehls

Stelle	1 - 3	Konstante BCE oder BCL
Stelle	4 - 5	CICS-Partition (hier: F7)
Stelle	7 - 10	Konstante INIT
Stelle	12 - 15	Transaktions-ID
Stelle	17 - 76	Daten, die übergeben werden sollen.

Oder bei LU6.2 Verbindung:

```
BCEF7 $CICP$ INIT TRAN Hier können Daten übergeben
                        werden.
```

CICP ist die SYSID eines mit LU6.2 verbundenen Systems.
Die Task wird dort gestartet.

Achtung: Ein CICS-Shutdown-Befehl kann nicht über eine Task abgesetzt, sondern nur über Konsolbefehle gestartet werden.

CICS-Task

Die Datenübergabe kann in der aufgerufenen Transaktion wie folgt gelöst werden:

CPG1 Beispiel:

```
H
I$CPG      KF
I
C                      READ $CPG          1  60 BATDAT
```

CPG2 Beispiel:

```
- OPTIONS
- -I.
-   FILE $CPG
-     1 60 BATDAT
- -C.
-   READ $CPG
```

COBOL Beispiel:

```
WORKING-STORAGE SECTION.
    02 BATDAT  PIC X(60).

PROCEDURE DIVISION.
    EXEC CICS RETRIEVE INTO (BATDAT) LENGTH (60)
```

CICS-Link

Mit BACEMT kann ein CICS-Command-Level-Programm vom Batch aus gestartet werden.

Während das CICS-Programm abläuft, wartet die Batch-Partition auf den erfolgreichen Ablauf dieses Programms.

Beispiel:

```
// JOB PROGRAMM
// EXEC BACEMT
BCEF7 LINK PROGRAMME Hier können Daten übergeben werden.
/*
/ &
```

Aufbau BACEMT-Befehl

Stelle	1 - 3	Konstante BCE oder BCL
Stelle	4 - 5	CICS-Partition (hier: F7)
Stelle	7 - 10	Konstante LINK
Stelle	12 - 19	Programmname
Stelle	21 - 80	Daten, die übergeben werden sollen.

BCEF7 \$CICP\$ LINK PROGRAMME. Hier können Daten übergeben werden.

Das Programm PROGRAMME wird im CICS mit der SYSID CICP gestartet. Wenn das Programm PROGRAMME beendet wird, kann über die Common Area der Return Code an Job Control übergeben werden.

CICS-Link

Die Datenübergabe kann in dem aufgerufenen Command-Level-Programm wie folgt gelöst werden:

CPG1 Beispiel:

```
      H
      ICPGCOM  F
      I
      C                SELCT                1 61 BATDAT
                                      CPGCOM
```

CPG2 Beispiel:

```
- OPTIONS
- -I.
-   FIELD CPGCOM
-       1 61 BATDAT
- -C.
-   SELECT CPGCOM
```

COBOL Beispiel:

```
LINKAGE SECTION.
01  DFHCOMMAREA.
    02  BATDAT  PIC X(61).

PROCEDURE DIVISION.
    IF EIBCALEN GREATER ZERO
    THEN
        ...
```

CICS-Link mit Übergabe des Returncodes

Bei der Datenübergabe kann auch ein Returncode mitgegeben werden, der dann über Conditional Job Control abgefragt werden kann.

CPG1 Beispiel:

```

1      H                                L8
      ICPGCOM  F
      I                                1  61  BATDAT
      C                                :
      C                                SELECT  CPGCOM
      C                                :
      C                                :
2      C                                EDIT  CPGCOM
      C                                :
3      OCPGCOM  F
3      O                                1X'00'
```

CPG2 Beispiel:

```

1      - OPTIONS MAIN COM ADD 8 PHASE TSTCOM.
      - I.
      -     FIELD CPGCOM
      -           1  61  BATDAT
      - C.
      -     :
      -     SELECT CPGCOM
      -     :
2      -     EDIT CPGCOM
      -     :
      - -O.
3      -     FIELD CPGCOM
3      -           1  '00'  HEX
```

Anmerkungen:

- zu 1: Als Adressierung muss 8 gesetzt werden. Das Programm arbeitet ohne TWA. Anderenfalls ist bei der Transaktion BACU eine TWASIZE von 3840 Bytes zu definieren.
- zu 2: Der Returncode muss vor Programmende mit der Operation EDIT CPGCOM gesetzt werden.
- zu 3: Der Returncode wird auf die Stelle 1 der Common Area gesetzt, und zwar in hexadezimaler Form.

BACEMT-Befehle als Parameter

BACEMT-Befehle können aus der Batch-Partition als Parameter abgesetzt werden.

Beispiel:

```
// JOB BATCH
// EXEC BACEMT,PARM='BCEF5 CEMT SET FILE(DATNAME) CLO'
/*
// EXEC UPDATE
/*
      :
// EXEC BACEMT,PARM='BCEF5 CEMT SET FILE(DATNAME) OPE'
/*
/&
```

Aufbau des BACEMT-Befehls

Der Aufbau des BACEMT-Befehls als Parameter entspricht dem Aufbau des üblichen Befehls.

BACEMT-Befehle als Parameter

BACEMT-Befehle können aus der Batch-Partition als Parameter mit Variablen abgesetzt werden.

Beispiel:

```
// JOB BATCH
// SETPARM XSTATF7=' '
// SETPARM PART=' '
// EXEC PROC=CPUVAR1,XSTATF7
// IF XSTATF7='ACTIVE' THEN
// GOTO A100
// GOTO A190                                KEIN OPEN DURCHFUEHREN
/. A100
// SETPARM PART='F7'
// EXEC BACEMT,PARM='BCE&PART CEMT SET FILE(DATNAME)OPEN'
/*
/. A190
/&
* $$ EOJ
```

Diese Funktion ermöglicht es, über die JCL Einfluss auf den BACEMT-Befehl zu nehmen.

Es kann z.B. zunächst der Status einer Partition abfragt und entsprechend dem Status eine Datei in dieser Partition geöffnet oder nicht geöffnet werden.

Zu beachten ist, dass die Variable vorher definiert wird.

Returncodes

BACEMT setzt den JCL-Returncode, der über Conditional Job Control abgefragt werden kann. Die Returncodes haben folgende Bedeutung:

- RC = 0 Alles OK, keine Fehler bei der Verarbeitung gefunden.
- RC = 4 Fehlerhafte Eingabe, CEMT Befehl hat falsche Syntax.
- RC = 8 CEMT entdeckte einen Fehler:
- entweder war ein Schlüsselwort falsch definiert oder
- es wurde ein CEMT Syntax Fehler erkannt oder
- fehlende Ressource angesprochen (Dateiname, Terminalname, Netname etc. falsch).
- RC = 12 Während der Verarbeitung erfolgte ein Programmabbruch des BACEMT-Programms.
- RC = 16 Es wurde ein XPCC-Fehler festgestellt:
- entweder wurde BACS START nicht gestartet oder
- die CICS-Partition ist nicht aktiv.
- RC = 20 Es wurde ein VSE-Fehler festgestellt.

Wird ab Stelle 73 eine Nummerierung über den Editor gesetzt, so werden die Stellen 73-80 auf Blank gelöscht.

XPCC-Returncodes

Als Reason Code wird ein zweiter Wert angezeigt, der von der Bedeutung her dem Returncode des XPCC-Befehls entspricht (IJBRETC).

IJBRETC = 00 Alles ok - keine Fehler gefunden.

IJBRETC = 04 BACEMT nicht gestartet, BACS START muss eingegeben werden, oder in der PLT muss BACSTA eingetragen sein.

IJBRETC = 06 XPCCB mit falschem Format, VSE nicht kompatibel.

IJBRETC = 08 Falscher Path-Id zugeordnet.

IJBRETC = 0E Kein Speicher in Power Partition für GETVIS.

IJBRETC = 0D Zu viele Connect Requests abgesetzt.

IJBRETC = 17 Power wird gerade beendet.

Alle weiteren Return Codes siehe Power Application Programming unter XPCC-Macro IJBXRETC.

Konsolmeldungen

Alle Programme von BACEMT senden Konsolmeldungen, wenn ein Fehler vorliegt. Die Meldungen haben folgende Bedeutung:

- MSGM01 BACEMT ATTACH SUCCESSFUL
BACEMT wurde online entweder über PLT oder über BACS START erfolgreich gestartet.
- MSGM02 xxxxxxxxx ERROR R15 = X'yyyyyyyy'
Während der Verarbeitung eines Befehles wurde ein Fehler festgestellt. Register 15 gibt Aufschluss über die Fehlerbedingung.
- MSGM03 ABEND XXXX AUFGETRETEN, BACS WURDE BEENDET
Es wurde ein CICS-Fehler festgestellt, das Programm BACSTA wurde abgebrochen, die Transaktion "BACS" hatte den Abendcode xxxx.
- MSGM04 UNBEKANNTER FEHLER AUFGETRETEN
Ein nicht identifizierbarer Fehler trat auf, BACS wird beendet.
- MSGM05 FALSCHER/FEHLENDER PARAMETER
Bei BACS START wurde nicht START eingegeben, BACS wird beendet, BACEMT ist nicht aktiv.
- MSGM06 BACEMT SHUTDOWN COMPLETE
BACEMT wurde normal beendet - entweder über BACS STOP oder über PLTSD Phase.
- MSGM07 TS QUEUE QIDERROR, BEI SHUTDOWN ABRUCH
Die interne Queue von BACEMT wurde bei Shutdown nicht gefunden.

Konsolmeldungen

- MSG01 XPCC xxxxxxxx ERROR IJBRETC=X'yy'.
Es wurde ein XPCC-Fehler festgestellt. Der Returncode wird in yy angezeigt.
- MSG02 XPCC xxxxxxxx SUCCESSFUL
Eine Verbindung war intern erfolgreich. Diese Meldung kommt nur bei Test.
- MSG03 BAC pp SHUTDOWN COMPLETE
Die Subtask im CICS wurde normal beendet.
- MSG04 Abnormal Termination. CODE=X'cc'.
In der Subtask BACSUB trat ein Programmfehler auf, der Code weist auf die Fehlerbedingung hin (sollte nicht auftreten).
- MSG05 Program Check X'xxxx' at offset +X'yyyyyy'.
Es wurde ein Programmfehler xxxx an Programmstelle yyyyyy entdeckt. Bitte diesen Fehler an uns melden.
- MSG06 BAC pp ABNORMAL TERMINATION.
Die Subtask wurde abnormal abgebrochen.
- MSG07 Requestor disconnected or terminated.
Diese Meldung tritt nach MSG04 - MSG06 auf. Die ursprüngliche Subtask ist nicht mehr verfügbar und nicht mit detach beendet worden.
- MSG08 CICS TASK ANTWORTET NICHT, CANCELED.
Es wurde ein CEMT-Befehl abgesetzt, der länger als 2 bzw. 8 Minuten dauert.

Dies kann auftreten, wenn ein CLOSE/OPEN auf alle Dateien gegeben wurde oder wenn eine Datei, die geschlossen wird, sich im Zugriff eines Dialogprogramms befindet. Es ist keine Rückmeldung an die VSE-Batch-Task möglich (beendet, mit RC 8).

Aktion: Nochmal mit BCL Command testen oder in mehrere Einzelbefehle aufteilen.

Konsolmeldungen

- MSG001 XPCC xxxxxxxx ERROR R15=X'xxxxxxxx' IJBRETC=X'yy'.
Es wurde ein XPCC-Fehler festgestellt. Der Returncode wird in yy angezeigt. Register 15 hat den Inhalt xxxxxxxx.
- MSG002 XPCC XXXXXXXX SUCCESSFUL
Erfolgreicher XPCC-Aufruf. Diese Meldung kommt nur im Testfall.
- MSG003 Abnormal Termination CODE =X'xx'
Es wurde ein Programmabbruch festgestellt, der Code xx gibt Aufschluss über den internen Fehler.
- MSG004 Programm Check X'xxxx' Offset +X'yyyyyyyy'
Es wurde ein Programmfehler festgestellt, der sich im BACEMT-Programm auf Stelle yyyyyyyy befindet.
- MSG005 VSE Error: XXXXXXXX R15 =X'yy'
Es wurde ein VSE-Fehler festgestellt, Register 15 enthält den VSE-Returncode.
- MSGC01 KEINE DATEN UEBERMITTELT, BACUPR BEENDET.
- MSGC02 ABEND XXXX AUFGETRETEN, BACUPR BEENDET.
- MSGC03 UNBEKANNTER FEHLER, BACUPR BEENDET.
- MSGC04 KEINE DATEN UEBERMITTELT, BACUPR BEENDET.
- MSGC05 BACUPR STARTUP NORMAL
- MSGC06 BACUPR SHUTDOWN NORMAL

DL/I OPEN - CLOSE

Um DL/I-Datenbanken zu eröffnen bzw. zu schließen ist bei BACEMT der Sourcecode eines Assemblerprogramms enthalten, der an die jeweilige Kundenumgebung angepasst werden muss. Dieser Sourcecode steht unter: BACDLOC.A

Die letzten beiden CSECTs beschreiben die Kundenvariablen und müssen angepasst werden. Die erste CSECT hat den Namen DBDTAB0 und enthält die DBD-Namen, die über das Programm BACDLOC gestartet oder beendet wird. Beispiel:

```
DBDTAB0  CSECT
          DC    CL16'DBDGEPa-DCNDKDPD'
          DC    CL16'DBDGEPa-DCNDKDSd'
          *-----*
          * Vor diesem Statement eigene Definitionen einfügen. *
          *-----*
          DC    CL16'---ENDE-----'
```

Die letzte CSECT heißt DLIVARS und enthält den SYSPCB-Namen und das Passwort. Diese Angaben sind vor der Umwandlung ebenfalls an die Kundengegebenheiten anzupassen.

Beispiel:

```
DLIVARS  CSECT
DLSYSPCB DC    CL8'SYSTEMDL'           <---anpassen
DLPASSWD DC    CL8'DLZPASS1'          <---anpassen
```

Beachte: In der CICS PPT muss das Programm BACDLOC eingetragen sein.

Der Aufruf erfolgt über LINK.

Beispiel:

```
// JOB BACDLI
// EXEC BACEMT
BCEF2 LINK BACDLOC  OPEN DBDGEPa
BCEF2 LINK BACDLOC  CLOS DBDLIEF
BCEF2 LINK BACDLOC  OPEN ALL
/*
```


DL/I OPEN - CLOSE

Folgende Returncodes werden gesetzt:

- RC=0 DLI-Verarbeitung erfolgreich.
- RC=8 OPEN/CLOS nicht durchgeführt, da der DB-Name nicht in der Tabelle DBDTAB0 stand.
- RC=12 Die Common Area war nicht 61 Stellen groß.
- RC=16 Die Stellen 1-4 in der Common Area waren nicht OPEN oder CLOS.