

# CICS-Sort

## **Kapitel 2: Funktion**

**2000**

Mit HMCSORT können im CICS Temporary Storage Queues sortiert werden.

Aus Performancegründen wird die TS-Queue jedoch im CICS nicht physisch umsortiert, sondern es wird nach dem Sort jeweils die nächste zu verarbeitende Satznummer der TS-Queue zur Verfügung gestellt. Die Daten werden dann im Anwendungsprogramm mit der ermittelten Satznummer gelesen.

## **Kapitel 3: Maximalwerte (erhöht ab Release 2.5)**

**3000**

Im CICS können beliebige Temporary-Storage-Bereiche sortiert werden. Mit einem bis zu 100-stelligen Sortierbegriff können alle Sätze in einer TS-Queue mit 32.767 Sätzen sortiert werden.

Nur wenn der Sortierbegriff länger ist als 100 Stellen, kann die Anzahl der sortierbaren Sätze kleiner als der Maximalwert von 32.767 sein. Anhand der gesamten Länge des Sortierbegriffs (KL) kann die maximale Anzahl der zu sortierenden Sätze (N) ermittelt werden:

$$N = 127 * 32100 / ( KL + 4 )$$

Der belegte Speicher (S) ergibt sich aus der Anzahl der Sätze und der Länge des Sortierbegriffs:

$$S = N * ( KL + 4 ) + 64.200$$

### **Achtung:**

Der CICS-Sort benutzt die erweiterte Storage Area (EDSA) der CICS-Partition. Diese sollte ausreichend hoch definiert sein, d.h. den belegten Speicher (S) \* Anzahl der möglichen parallel aktiven Sort-Tasks (T) aufnehmen können:

$$\text{EDSA für CICS-Sort} = S * T$$

Ist der Speicher zu klein definiert, kann es zur Situation „Short on Storage“ kommen. Kann das CICS keinen Speicher mehr zur Verfügung stellen, dann wird der CICS-Sort mit dem Returncode 1 abgebrochen.

Es muss CPG installiert sein, Version 2.5 oder höher.

In der PPT muss die Phase HMCSORT (Typ Assembler) eingetragen worden sein.

Das Modul CSORT muss in der allgemeinen HL1-Tabelle angefügt werden. Die HL1-Tabelle muss neu katalogisiert worden sein.

Im Data Dictionary muss die Datei CSORT (Input, Fix, Satzlänge 256, Einheit HL1DS) angelegt worden sein.

Beispielprogramm

8000

OPTIONS DATASET

FILE CSORT.

FILE QTFS.

-C

BLANK 2.

DRID 4.

MSG 40.

PAGE 3 0.

QNR 5 0.

S 3 0.

PARMS 3 \* 8.

-I

FILE QTFS

1 80 SATZ.

FILE CSORT HS DD

1 8 CPGTSN.

PAC 9 10 0 RC.

11 14 CPGHIC.

PAC 15 17 0 QNR.

PAC 18 19 0 S.

PAC 21 44 PARMS.

-C.

:

:

CPGTSN = CPGTID.

MOVE 'QTFS' TO CPGTSN.

PARMS(1) = '000302CA'.

PARMS(2) = '000101CD'.

S = 2.

SETLL CSORT.

\* Dataset zum Sortieren

\* Benutzer-Storage

\* Drucker-Id für das LIST

\* Fehlermeldung

\* Seite in der Liste

\* Satznr. des sortierten Bereichs

\* Anzahl Sortierparameter

\* muß immer gesetzt werden !!!

\* 3 Sortierparameter in der

\* Form SSSSLLTA:

\* SSSS Startposition

\* LL in der Länge

\* Typ C = Character (alpha)

\* P = gepackt

\* Art A/D (auf-/absteigend)

\* Daten des Benutzer-Storage

\* variabler Temporary Storage-

\* Name für die Übergabe des

\* Benutzer-Storage-Namens

\* Return Code

\* HL1 Interface Control Field

\* Satznummer der Queue

\* Anzahl Sortierparameter

\* Sortierparameter (bis zu 3)

\* -----

\* Die TS-Queue Terminal-Id/QTFS

\* ist erstellt und wird hier

\* nach 2 Kriterien sortiert

\* -----

\* Name der zu sortierenden TS-Q

\* von Stelle 3 in der Länge 2

\* nach Charactern aufsteigend

\* von Stelle 1 in der Länge 1

\* nach Charactern absteigend

\* 2 Sortierparameter benutzt

\*

\*

\* Sortieren nach Parametern

```
IF RC = 0.
```

\* Nach dem Sortieren kontrol-  
\* liert man den Return Code  
\* Alles ok, dann werden die  
\* Daten sortiert verarbeitet,  
\* z.B. als Liste ausgegeben.  
\* -----

```
    DRID LIST LISTE HEADER.  
    DO WHILE RC = 0 AND.  
    WHILE CPGFRC = BLANK.  
        READ CSORT.  
        IF CPGFRC = BLANK.  
            QNR READ QTFS.  
            IF CPGFRC = BLANK.  
                DRID LIST LISTE DETAIL.  
            ENDIF  
        ENDIF  
    ENDDO
```

\* Solange Sätze da sind  
\* **Ermittle QNR für nächsten Satz**  
\* End of File = Schleifenende  
\* **Lese nächsten Satz der Sor-**  
\* **tierreihenfolge**  
\* End of File = Schleifenende  
\* Drucke den Satz

```
    RNDOM CSORT.
```

\* -----  
\* **Sortierbereich freigeben, wenn**  
\* **er nicht mehr benötigt wird**  
\* -----

```
    DRID LIST LISTE TRAILR  
END
```

\* Fehlerabfragen

```
EVALUATE.  
    WHEN RC = 1  
        MSG = 'SORTAREA IS FULL      '.  
    WHEN RC = 2  
        MSG = 'INVALID SEQUENCE DEF.'  
    WHEN RC = -1  
        MSG = 'END OF TESTVERSION  '.  
END-EVALUATE
```