

SORTKEY - Sort-Key Function

SORTKEY (*character-string*)

Diese Systemfunktion dient zum Konvertieren von "nicht richtig sortierten" Zeichen (oder Kombinationen von Zeichen) in andere Zeichen (oder Kombinationen von Zeichen), die vom Sortierprogramm oder Datenbanksystem alphabetisch "richtig sortiert" werden.

Format/Länge:

A253

Es gibt in vielen Landessprachen Zeichen (oder Zeichenkombinationen), die von einem Sortierprogramm oder Datenbanksystem nicht in der richtigen alphabetischen Reihenfolge sortiert werden, da die Reihenfolge der Zeichen im vom Computer verwendeten Zeichensatz nicht immer der alphabetischen Reihenfolge der Zeichen entspricht.

Zum Beispiel wird der spanische Buchstabe CH in der Regel von einem Sortierprogramm bzw. Datenbanksystem wie zwei Buchstaben behandelt und zwischen CG und CI einsortiert — gehört aber eigentlich als eigener Buchstabe im spanischen Alphabet zwischen "C" und "D".

Oder es kann sein, dass Kleinbuchstaben und Großbuchstaben entgegen Ihren Wünschen bei der Sortierreihenfolge nicht gleich behandelt werden, dass Ziffern vor Buchstaben sortiert werden (Sie aber wünschen, dass Buchstaben vor Ziffern sortiert werden) oder dass Sonderzeichen (z.B. Bindestriche in Doppelnamen) zu einer unerwünschten Sortierreihenfolge führen.

In solchen Fällen können Sie die Systemfunktion `SORTKEY(character-string)` benutzen. Die von `SORTKEY` berechneten Werte werden nur als Sortierkriterium benutzt, während die ursprünglichen Werte für die Interaktion mit dem Endbenutzer verwendet werden.

Sie können die `SORTKEY`-Funktion in einem `COMPUTE` sowie in einer logischen Bedingung als arithmetischen Operanden verwenden.

Als *character-string* können Sie eine alphanumerische Konstante oder Variable oder eine einzelne Ausprägung eines alphanumerischen Arrays angeben.

Wenn Sie die `SORTKEY`-Funktion in einem Natural-Programm angeben, wird der User-Exit `NATUSKnn` aufgerufen — wobei *nn* der aktuelle Sprachcode (d.h. der aktuelle Wert der Systemvariablen `*LANGUAGE`) ist.

Sie können diesen User-Exit in jeder Programmiersprache, die über eine Standard-CALL-Schnittstelle verfügt, schreiben. Der mit `SORTKEY` angegebene *character-string* wird an den User-Exit übergeben. Der User-Exit muss so programmiert werden, dass er "falsch sortierte" Zeichen in dieser Zeichenkette in entsprechende "richtig sortierte" Zeichen umsetzt. Die umgesetzte Zeichenkette wird dann vom Natural-Programm zur weiteren Verarbeitung verwendet.

Allgemeine Aufruf-Konventionen für externe Programme sind in der Dokumentation zum `CALL`-Statement erläutert.

Nähere Informationen zu den Aufruf-Konventionen für `SORTKEY` User-Exits finden Sie beim `CALL`-Statement in *User Exits* in der *Natural Statements*-Dokumentation.

Beispiel:

```

DEFINE DATA LOCAL
1 CUST VIEW OF CUSTOMERFILE
  2 NAME
  2 SORTNAME
END-DEFINE
...
*LANGUAGE := 4
...
REPEAT
  INPUT NAME
  SORTNAME := SORTKEY(NAME)
  STORE CUST
  END TRANSACTION
  ...
END-REPEAT
...
READ CUST BY SORTNAME
  DISPLAY NAME
END-READ
...

```

Angenommen, im obigen Beispiel würden bei mehrmaliger Ausführung des INPUT-Statements nacheinander die Werte "Sanchez", "Sandino" und "Sancinto" eingegeben.

Bei der Zuweisung von SORTKEY(NAME) zu SORTNAME würde der User-Exit NATUSK04 aufgerufen. Dieser müsste so programmiert werden, dass er zunächst alle Kleinbuchstaben in Großbuchstaben umsetzt und dann die Zeichenfolge "CH" in "Cx" umsetzt — wobei x dem letzten Zeichen im verwendeten Zeichensatz entspricht, also hexadezimal H' FF' (vorausgesetzt dieses letzte Zeichen ist kein druckbares Zeichen).

Es werden sowohl die "eentlichen" Namen (NAME) als auch die für die gewünschte Sortierung umgesetzten Namen (SORTNAME) gespeichert. Zum Lesen der Datei wird SORTNAME verwendet. Dann würden bei Ausführung des DISPLAY-Statements die Namen in der richtigen spanischen alphabetischen Reihenfolge ausgegeben:

```

Sancinto
Sanchez
Sandino

```