

WRITE WORK FILE

<code>WRITE WORK [FILE] work-file-number [VARIABLE] operand1 ...</code>

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Externe Darstellung der Felder
- Verarbeitung großer und dynamischer Variablen
- Beispiel

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Verwandte Statements: DEFINE WORK FILE | READ WORK FILE | CLOSE WORK FILE |
DOWNLOAD PC FILE

Gehört zur Funktionsgruppe: *Verarbeitung von Arbeitsdateien/PC-Dateien*

Funktion

Das Statement `WRITE WORK FILE` dient dazu, Datensätze auf eine physisch-sequentielle Arbeitsdatei (Work File) zu schreiben.

Auf Großrechnern kann dieses Statement nur im Batch-Betrieb oder unter Complete, CMS, TSO und TIAM verwendet werden. Entsprechend JCL- oder Systemkommandos müssen ausgeführt werden, um die Arbeitsdatei zuzuweisen. Weitere Informationen siehe *Operations*-Dokumentation. Siehe auch Profilparameter WORK in der *Parameter-Referenz*.

Es ist möglich, in einem Programm oder einer Verarbeitungsschleife eine Arbeitsdatei zu erstellen und diese dann in einem anderen Programm oder einer anderen eigenständigen Verarbeitungsschleife mit einem `READ WORK FILE`-Statement zu lesen.

Anmerkung:

Bezüglich Unicode-Support siehe *Work Files and Print Files on Mainframe Platforms* in der *Unicode and Code Page Support*-Dokumentation.

Syntax-Beschreibung

Operanden-Definitionstabelle:

Operand	Mögliche Struktur				Mögliche Formate												Referenzierung erlaubt	Dynam. Definition
<i>operand1</i>	C	S	A	G	A	U	N	P	I	F	B	D	T	L	C	G	ja	nein

Anmerkung:

Bei Natural Connection gilt weder Format C noch Format G.

Syntax-Element-Beschreibung:

<i>work-file-number</i>	<p>Arbeitsdateinummer:</p> <p>Die für Natural definierte Nummer der Arbeitsdatei, die verwendet werden soll.</p>
VARIABLE	<p>Variablen-Eintrag:</p> <p>Es ist möglich, mittels verschiedener WRITE WORK FILE-Statements Datensätze mit verschiedenen Feldern auf dieselbe Arbeitsdatei zu schreiben. In diesem Fall müssen alle betreffenden WRITE WORK FILE-Statements das Schlüsselwort VARIABLE enthalten; die Datensätze werden dann mit variablem Format auf die externe Datei geschrieben. Natural schreibt alle Ausgabedateien in variablen Blöcken (es sei denn, Sie geben in der Ausführungs-JCL Datensatzformat und Blockgröße an).</p> <p>Wenn die Operanden-Liste eine dynamische Variable enthält (die je nach Ausführungsart des WRITE WORK FILE-Statements eine andere Größe annimmt), muss der VARIABLE-Eintrag in allen WRITE WORK FILE-Statements angegeben werden.</p> <p>Variabler Indexbereich:</p> <p>Wenn Sie ein Array auf eine Arbeitsdatei schreiben, können Sie für das Array einen variablen Indexbereich angeben. Zum Beispiel:</p> <pre>WRITE WORK FILE work-file-number VARIABLE #ARRAY (I:J)</pre>
<i>operand1</i>	<p>Felder:</p> <p>Als <i>operand1</i> geben Sie die Felder an, die auf die Arbeitsdatei geschrieben werden sollen. Dies können entweder Datenbankfelder, Benutzervariablen und/oder Felder sein, die mit einem READ WORK FILE-Statement von einer anderen Arbeitsdatei gelesen wurden.</p> <p>Bei Datenbank-Arrays kann durch eine einen Bereich umfassende Indexierung angegeben werden, welche Ausprägungen auf die Arbeitsdatei geschrieben werden sollen. Feldgruppen können durch Angabe des Gruppennamens referenziert werden; alle Felder einer Gruppe werden einzeln auf die Arbeitsdatei geschrieben.</p>

Externe Darstellung der Felder

Mit einem WRITE WORK FILE-Statement auf eine Arbeitsdatei geschriebene Felder werden auf der externen Datei entsprechend ihrer internen Definition dargestellt. Die Feldwerte werden nicht verändert.

Bei Feldern der Formate A oder B entspricht die Anzahl der Bytes auf der externen Datei der programminternen Längendefinition. Die Feldwerte werden nicht verändert; ein Komma (Dezimalpunkt) wird nicht wiedergegeben.

Bei Feldern des Formats N ergibt sich die Anzahl der Bytes auf der externen Datei aus der Summe der Stellen vor und nach dem Komma. Das Komma (Dezimalpunkt) wird auf der externen Datei nicht wiedergegeben.

Bei Feldern des Formats P ergibt sich die Anzahl der Bytes auf der externen Datei aus der Summe der Stellen vor und nach dem Komma plus einer Stelle für das Vorzeichen, geteilt durch 2, wobei auf ganze Bytes aufgerundet wird.

Anmerkung:

Beim Schreiben von Feldern auf eine Arbeitsdatei erfolgt keine Umsetzung von Feldformaten.

Beispiele für Felddarstellung:

Felddefinition	Ausgabelänge
#FIELD1 (A10)	10 Bytes
#FIELD2 (B15)	15 Bytes
#FIELD3 (N1.3)	4 Bytes
#FIELD4 (N0.7)	7 Bytes
#FIELD5 (P1.2)	2 Bytes
#FIELD6 (P6.0)	4 Bytes

Anmerkung:

Wenn die Systemfunktionen AVER, NAVER, SUM oder TOTAL für numerische Felder (Format N oder P) auf eine Arbeitsdatei geschrieben werden, vergrößert sich intern die Länge dieser Felder um eine Stelle (z.B.: SUM eines Feldes vom Format P3 wird auf P4 verlängert). Dies ist beim Lesen der Arbeitsdatei zu berücksichtigen.

Verarbeitung großer und dynamischer Variablen

Arbeitsdateityp	Verarbeitung
UNFORMATTED	Auf Großrechnern kann der Arbeitsdateityp UNFORMATTED zum Schreiben von Variablen benutzt werden, deren Länge die maximale Datensatzlänge überschreitet. Siehe auch den Abschnitt <i>Zugriff auf Arbeitsdateien mit großen und dynamischen Variablen</i> .
FORMATTED	Eine dynamische Variable wird in ihrer aktuell definierten Länge (einschließlich Länge 0) geschrieben.

Beispiel

```
** Example 'WWFEX1': WRITE WORK FILE
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
END-DEFINE
*
FIND EMPLOY-VIEW WITH CITY = 'LONDON'
  WRITE WORK FILE 1
    PERSONNEL-ID NAME
END-FIND
*
END
```