

DEFINE WORK FILE

$\text{DEFINE WORK FILE } n \left\{ \begin{array}{l} \textit{operand1} [\text{TYPE } \textit{operand2}] \\ \text{TYPE } \textit{operand2} \end{array} \right\} [\text{ATTRIBUTES } \{\textit{operand3}\} \dots]$
--

Anmerkung:

Die in eckigen Klammern [...] gezeigten Elemente sind optional, aber mindestens eines muss bei diesem Statement angegeben werden.

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Arbeitsdateiname unter z/OS Batch, TSO und Server
- Arbeitsdateiname unter z/VSE Batch
- Arbeitsdateiname unter VM/CMS
- Arbeitsdateiname unter BS2000/OSD Batch und TIAM
- Arbeitsdateiname unter CICS
- Arbeitsdateiname unter Com-plete/SMARTS

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

Verwandte Statements: CLOSE WORK FILE | READ WORK FILE | WRITE WORK FILE

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

Funktion

Das Statement `DEFINE WORK FILE` dient dazu, innerhalb einer Natural-Anwendung einer Natural-Arbeitsdateinummer einen Dateinamen zuzuweisen.

Damit können Sie Arbeitsdatei-Zuweisungen innerhalb Ihrer Natural-Session dynamisch vornehmen bzw. ändern, sowie auf anderer Ebene gemachte Arbeitsdatei-Zuweisungen überschreiben.

Ist bei der Ausführung dieses Statements die angegebene Arbeitsdatei bereits offen, bewirkt dieses Statement implizit, dass die Arbeitsdatei geschlossen wird.

Alle während einer Session zu benutzenden Arbeitsdateien müssen mittels des Subparameters `AM` des Profilparameters `WORK` oder automatisch durch Definition in der JCL im Voraus einer Zugriffsmethode zugewiesen werden.

<i>operand1</i>	<p><i>operand1</i> ist der Name der Arbeitsdatei.</p> <p>Als <i>operand1</i> geben Sie den Namen des der Arbeitsdateinummer zuzuweisenden Datasets an. <i>operand1</i> kann 1 bis 253 Stellen lang sein. Sie können entweder einen logischen oder einen physischen Dateinamen angeben. Das mögliche Format ist abhängig von der Betriebssystemumgebung und der vom Subparameter AM des Profilparameters WORK definierten Zugriffsmethode. Einige Zugriffsmethoden unterstützen <i>operand1</i> nicht als Arbeitsdateinamen, z.B. AM=COMP und AM=PC.</p> <p>Wenn Sie <i>operand1</i> weglassen, wird der Wert von <i>operand1</i> bestimmt, indem für diese Arbeitsdateinummer der aktuelle Name genommen wird, der bei dem zuvor ausgeführten DEFINE WORK FILE-Statement angegeben wurde. Wenn noch kein DEFINE WORK FILE-Statement ausgeführt worden ist, wird der im Natural-Parametermodul NATPARM angegebene Name verwendet.</p> <p>Anmerkung: Wenn <i>operand1</i> nicht angegeben wird, ist das Verhalten bei Natural für Großrechner und Natural für Windows/UNIX/OpenVMS verschieden.</p> <p>Informationen zu Betriebssystem- oder TP-Monitor-abhängigen Arbeitsdateinamen finden Sie in den folgenden Abschnitten:</p> <ul style="list-style-type: none">● Arbeitsdateiname unter z/OS Batch, TSO und Server● Arbeitsdateiname unter z/VSE Batch● Arbeitsdateiname unter VM/CMS● Arbeitsdateiname unter BS2000/OSD Batch und TIAM● Arbeitsdateiname unter CICS● Arbeitsdateiname unter Com-plete/SMARTS
-----------------	--

TYPE <i>operand2</i>	<i>operand2</i> gibt den Arbeitsdateityp an. Beim Wert von <i>operand2</i> wird nicht nach Groß- und Kleinbuchstaben unterschieden. Der Wert muss in Anführungszeichen (' . . . ') stehen oder in einer alphanumerischen Variable angegeben werden.	
	UNFORMATTED	Eine vollkommen unformatierte Datei. Es werden keine Formatier-Informationen geschrieben (weder für Felder noch für Sätze). UNFORMATTED behandelt eine Arbeitsdatei als einen Byte-Strom ohne Datensatz-Grenzen. Beachten Sie, dass der Type UNFORMATTED von Entire Connection zurückgewiesen wird. Format: UNFORMATTED
	FORMATTED	FORMATTED definiert eine reguläre datensatz-orientierte Arbeitsdatei, die derselben Verarbeitung unterliegt wie in vorherigen Natural-Versionen.
[ATTRIBUTES {<i>operand3</i>}...]	ATTRIBUTES-Klausel: Diese Klausel ist nur in Natural für Open Systems sinnvoll; in Natural für Großrechner wird sie ignoriert.	

Beispiele:

```
DEFINE WORK FILE 17 #FILE TYPE 'UNFORMATTED'
#TYPE := 'FORMATTED'
DEFINE WORK FILE 18 #FILE TYPE #TYPE
```

Arbeitsdateiname unter z/OS Batch, TSO und Server

Folgende Themen werden behandelt:

- Arbeitsdateiname - operand1
- Zuweisung und Freigabe von Datasets
- Arbeitsdateien in Server-Umgebungen

Arbeitsdateiname - operand1

Unter z/OS kann für eine Arbeitsdateinummer, die mit Zugriffsmethode AM=STD definiert ist, *operand1* folgendes sein:

- ein logischer Dataset-Name (DD-Name, 1 bis 8 Stellen);
- ein physischer Dataset-Name eines katalogisierten Datasets (1 bis 44 Stellen) oder ein physischer Dataset-Member-Name;

- ein Pfad- und Member-Name einer HFS-Datei (1 bis 253 Stellen) in einer MVS-UNIX-Services-Umgebung;
- eine JES-Spoolfile-Klasse;
- NULLFILE.

Logische Dataset-Namen	<p>Beispiel:</p> <pre>DEFINE WORK FILE 21 'SYSOUT1'</pre> <p>Das angegebene Dataset SYSOUT1 muss zugewiesen worden sein, bevor das DEFINE WORK FILE-Statement ausgeführt wird.</p> <p>Die Zuweisung kann über JCL, CLIST (TCO) oder dynamische Zuweisung (SVC 99) erfolgen. Für dynamische Zuweisung können Sie die Programmierschnittstelle (API) USR2021N in der Library SYSEXT verwenden.</p> <p>Der im DEFINE WORK FILE-Statement angegebene Dataset-Name überschreibt den im Subparameter DEST des Profilparameters WORK angegebenen Namen.</p> <p>Optional kann dem Dataset-Namen DDN= vorangestellt werden, um anzuzeigen, dass es sich um einen DD-Namen handelt. Zum Beispiel:</p> <pre>DEFINE WORK FILE 22 'DDN=MYWORK'</pre>
Physische Dataset-Namen	<p>Beispiel:</p> <pre>DEFINE WORK FILE 23 'TEST.WORK.FILE'</pre> <p>Das angegebene Dataset muss in katalogisierter Form vorhanden sein. Wenn das DEFINE WORK FILE-Statement ausgeführt wird, wird das Dataset dynamisch über SVC 99 mit dem aktuellen DD-Namen und der Option DISP=SHR zugewiesen.</p> <p>Wenn der Dataset-Name 8 Stellen oder kürzer ist und keinen Punkt (.) enthält, könnte er fälschlich als DD-Name interpretiert werden. Um dies zu vermeiden, stellen Sie ihm DSN= voran. Zum Beispiel:</p> <pre>DEFINE WORK FILE 22 'DSN=WORKXYZ'</pre> <p>Falls das Dataset ein PDS-Member ist, geben Sie den PDS-Member-Namen (1 bis 8 Stellen) in Klammern hinter dem Dataset-Namen (1 bis 44 Stellen) an. Zum Beispiel:</p> <pre>DEFINE WORK FILE 4 'TEST.WORK.PDS(TEST1)'</pre> <p>Falls das angegebene Member nicht existiert, wird ein neues Member unter diesem Namen angelegt.</p>

<p>HFS-Dateien</p>	<p>Beispiel:</p> <pre>DEFINE WORK FILE 14 '/u/nat/rec/test.txt'</pre> <p>Der angegebene Pfadname muss existieren. Wenn das DEFINE WORK FILE-Statement ausgeführt wird, wird die HFS-Datei dynamisch zugewiesen. Falls das angegebene Member nicht existiert, wird ein neues Member unter diesem Namen angelegt.</p> <p>Bei der dynamischen Zuweisung des Datasets werden folgende z/OS-Pfadoptioen verwendet:</p> <pre>PATHOPTS=(OCREAT , OTRUNC , ORDWR) PATHMODE=(SIRUSR , SIWUSR , SIRGRP , SIWGRP) FILEDATA=TEXT</pre> <p>Wird eine HFS-Datei geschlossen, wird sie automatisch von z/OS freigegeben (unabhängig vom Wert des Subparameters FREE des Profilparameters WORK).</p> <p>Um eine HFS-Datei zu lesen, müssen Sie statt des DEFINE WORK FILE-Statements die Programmierschnittstelle (API) USR2021N (dynamische Dataset-Zuweisung) verwenden, und zwar wegen der OTRUNC-Option. Diese Option setzt die HFS-Datei beim ersten Lesezugriff zurück und bewirkt eine leere Datei.</p>
<p>JES-Spoolfile-Klasse</p>	<p>Um ein JES-Spool-Dataset zu erzeugen, geben Sie SYSOUT=x an (wobei x die gewünschte Spoolfile-Klasse ist). Für die Standard-Spoolfile-Klasse geben Sie SYSOUT=* an.</p> <p>Beispiele:</p> <pre>DEFINE WORK FILE 10 'SYSOUT=A' DEFINE WORK FILE 12 'SYSOUT=*'</pre> <p>Um zusätzliche Parameter für die dynamische Zuweisung anzugeben, verwenden Sie statt des DEFINE WORK FILE-Statements die Programmierschnittstelle (API) USR2021N (dynamische Dataset-Zuweisung) in der Library SYSEXT.</p>
<p>NULLFILE</p>	<p>Um ein Dummy-Dataset zuzuweisen.</p>

Zuweisung und Freigabe von Datasets

Wenn das DEFINE WORK FILE-Statement ausgeführt wird und ein physischer Dataset-Name, eine HFS-Datei, eine Spoolfile-Klasse oder ein Dummy-Dataset angegeben wurde, wird das entsprechende Dataset automatisch zugewiesen. Wenn die logische Datei bereits geöffnet ist, wird sie automatisch geschlossen, außer wenn der Subparameter CLOSE=FIN des Profilparameters WORK angegeben wurde, wobei dann eine Fehlermeldung ausgegeben wird. Außerdem wird ein bestehendes Dataset mit dem gleichen aktuellen DD-Namen automatisch freigegeben, bevor das neue Dataset zugewiesen wird. Um Fehler durch verfrühtes Öffnen von beim Programmstart noch nicht zugewiesenen Arbeitsdateien zu vermeiden, sollten Arbeitsdateien mit dem Subparameter OPEN=ACC (Öffnen bei erstem Zugriff) im Profilparameter WORK definiert werden.

Im Falle einer HFS-Datei oder einer im Profilparameter `WORK` mit Subparameter `FREE=ON` definierten Arbeitsdatei wird die Arbeitsdatei automatisch freigegeben, sobald sie geschlossen worden ist. Als Alternative steht Ihnen für die dynamische Zuweisung und Freigabe von Datasets die Programmierschnittstelle (API) `USR2021N` in Library `SYSEXT` zur Verfügung. Dieses API ermöglicht auch die Angabe zusätzlicher Parameter für die dynamische Zuweisung.

Arbeitsdateien in Server-Umgebungen

In Server-Umgebungen kann es zu Fehlern kommen, wenn mehrere Natural-Sessions versuchen, ein Dataset mit dem gleichen DD-Namen zuzuweisen oder zu öffnen. Um dies zu vermeiden, geben Sie entweder im Profilparameter `WORK` den Subparameter `DEST=*` an, oder Sie programmieren `DEFINE WORK FILE '*'` vor dem eigentlichen `DEFINE WORK FILE`-Statement; Natural generiert dann einen eindeutigen DD-Namen bei der Zuweisung der physischen Datasets, wenn das erste `DEFINE WORK FILE`-Statement für die betreffende Arbeitsdatei ausgeführt wird.

Alle Arbeitsdateien, deren DD-Namen mit `CM` anfangen, werden von allen Sessions in einer Server-Umgebung gemeinsam benutzt. Eine gemeinsame benutzte Arbeitsdatei, die für Ausgabe von der ersten Session geöffnet wird, wird bei Beendigung des Servers physisch geschlossen. Eine gemeinsame benutzte Arbeitsdatei, die für Eingabe geöffnet wird, wird physisch geschlossen, wenn die letzte Session sie schließt, d.h. wenn sie eine Dateiende-Bedingung erhält. Beim gleichzeitigen Lesen einer gemeinsam benutzten Arbeitsdatei wird einem `READ WORK FILE`-Statement nur ein Datei-Datensatz geliefert.

Arbeitsdateiname unter z/VSE Batch

Unter `z/VSE` kann *operand1* für eine mit `AM=STD` definierte Arbeitsdatei Folgendes sein:

- ein logischer Dataset-Name (DD-Name, 1 bis 7 Zeichen)
- `NULLFILE` (um ein Dummy-Dataset anzugeben).

Logische Dataset- Namen	<p>Beispiel:</p> <pre>DEFINE WORK FILE 21 'SYSOUT1'</pre> <p>Das angegebene Dataset <code>SYSOUT1</code> muss in der JCL oder in den <code>z/VSE</code> Standard bzw. Partition Labels definiert worden sein.</p> <p>Der im <code>DEFINE WORK FILE</code>-Statement angegebene Dataset-Name überschreibt den im Subparameter <code>DEST</code> des Profilparameters <code>WORK</code> angegebenen Namen.</p> <p>Optional kann dem Dataset-Namen <code>DDN=</code> vorangestellt werden, um anzuzeigen, dass es sich um einen DD-Namen handelt. Zum Beispiel:</p> <pre>DEFINE WORK FILE 22 'DDN=MYWORK'</pre>
NULLFILE	<p>Um ein Dummy-Dataset zuzuweisen, geben Sie <code>NULLFILE</code> als <i>operand1</i> an:</p> <pre>DEFINE WORK FILE n 'NULLFILE'</pre>

Arbeitsdateiname unter VM/CMS

Unter VM/CMS gilt (für im Natural-Parametermodul mit AM=STD definierte Arbeitsdateien) dasselbe für *operand1* wie unter z/OS (siehe oben), aber mit folgenden Unterschieden:

- Anstatt dynamischer Zuweisung über MVS SVC 99 wird das CMS-Kommando FILEDEF verwendet, um eine Datei zu definieren.
- HFS-Dateien werden nicht unterstützt.
- JES Spool-Klassen werden nicht unterstützt.
- Außerdem wird die folgende Syntax verwendet:

```
DEFINE WORK FILE n 'fname ftype fmode (options)'
```

Folgendes CMS-Kommando wird generiert:

```
FILEDEF ddname-n DISK fname ftype fmode (options)
```

- Außerdem ist die folgende Syntax zulässig:

```
DEFINE WORK FILE n 'FILEDEF=filedef-parameters'
```

Folgendes CMS-Kommando wird generiert:

```
FILEDEF ddname-n =filedef-parameters
```

Zum Beispiel:

```
DEFINE WORK FILE 5 'FILEDEF=TAP1 SL 2 VOLID BKUP08 (BLKSIZE 20000)'
```

Folgendes CMS-Kommando wird generiert:

```
FILEDEF CMWKF05 TAP1 SL 2 VOLID BKUP08
```

Für eine mit AM=CMS definierte Arbeitsdatei kann DEFINE WORK FILE zum Ändern des Ausgabemediums benutzt werden. Zusätzlich zu den Ausgabemedien, die mit dem Subparameter DEST im Parameter-Makro NETWORK angegeben werden können, kann ein REXX-Stem definiert werden. Einzelheiten entnehmen Sie dem Abschnitt *Print File and Work File Support* in der *Operations*-Dokumentation.

Arbeitsdateiname unter BS2000/OSD Batch und TIAM

Unter BS2000/OSD können Sie für eine Arbeitsdateinummer, die mit Zugriffsmethode AM=STD definiert ist, mit *operand1* einen Dateinamen oder Link-Namen angeben, der dieser Arbeitsdatei zugewiesen werden soll.

In diesem Fall kann *operand1* eine Länge von 1 bis 253 Zeichen haben und eine der folgenden Bedeutungen haben:

- einen BS2000/OSD Link-Namen (1 bis 8 Zeichen)

- einen BS2000/OSD Dateinamen (9 bis 54 Zeichen)
- einen generischen BS2000/OSD Dateinamen (Wildcard)
- einen BS2000/OSD Dateinamen und Link-Namen
- einen generischen BS2000/OSD Dateinamen und Link-Namen (Wildcard)
- *DUMMY

Es gelten die folgenden Regeln:

1. Dateiname und Link-Name können als Positionsparameter oder Schlüsselwort-Parameter angegeben werden. Die entsprechenden Schlüsselwörter sind FILE= und LINK=. Das Vermischen von Positionsparameter und Schlüsselwort-Parameter ist zulässig aber nicht empfehlenswert.
2. Eine Zeichenkette mit einer Länge von 1 bis 8 Zeichen ohne Kommas wird als ein Link-Name interpretiert. Diese Notation ist mit früheren Versionen von Natural kompatibel.

```
DEFINE WORK FILE 1 'W01'
```

Die entsprechende Definition lautet mit Schlüsselwort-Parameter:

```
DEFINE WORK FILE 1 'LINK=W01'
```

3. Eine Zeichenkette von 9 bis 54 Zeichen ohne Kommas wird als ein Dateiname interpretiert.

Beispiel:

```
DEFINE WORK FILE 2 'NATURALvr.TEST.WORKFILE02'
```

dabei steht *vr* für die Natural-Version und -Release-Nummer.

Die entsprechende Definition lautet mit Schlüsselwort-Parameter:

```
DEFINE WORK FILE 2 'FILE=NATURALvr.TEST.WORKFILE02'
```

4. Die folgende Eingabe wird interpretiert, ohne die Länge zu berücksichtigen und bildet deshalb eine Ausnahme von den Regeln 2 und 3:

- Schlüsselwort-Eingabe: LINK=, FILE=
- *DUMMY
- NULLFILE (entspricht *DUMMY)
- *
- *, *

Beispiel: DEFINE WORK FILE 7 'FILE=Y' ist eine gültige Dateizuweisung und kein Link-Name, obwohl die Zeichenkette weniger als 9 Zeichen enthält.

5. Generische Dateinamen sind wie folgt aufgebaut:

Wnn.userid.tsn.date.time.number

dabei ist

<i>nn</i>	eine Arbeitsdateinummer
<i>userid</i>	eine Natural User-ID mit 8 Zeichen
<i>tsn</i>	die BS2000/OSD TSN der aktuellen Task mit 4 Ziffern
<i>date</i>	<i>DDMMYYYY</i> (<i>DD</i> = Tag, <i>MM</i> = Monat, <i>YYYY</i> = Jahr)
<i>time</i>	<i>HHIISS</i> (<i>HH</i> = Stunde, <i>II</i> = Minute, <i>SS</i> = Sekunde)
<i>number</i>	eine Zahl mit 5 Ziffern

6. Generische Link-Namen sind wie folgt aufgebaut:

NWFnnnnn

nnnnn ist eine 5-stellige Zahl, die nach jeder Generierung eines dynamischen Link-Namens um eins erhöht wird.

7. Das Ändern der Dateizuweisung für eine Arbeitsdateinummer führt zu einem impliziten Schließen (CLOSE) der bisher zugewiesenen Arbeitsdatei.

Es empfiehlt sich in allen Fällen, außer wenn Sie einen Link-Namen (zum Beispiel: W01) angeben, mit Schlüsselwort-Parametern zu arbeiten. Dies hilft Konflikte bei der Interpretation zusätzlicher Reports zu vermeiden und ist von Bedeutung für Dateinamen mit weniger als 9 Zeichen.

Beispiel:

```
DEFINE WORK FILE 3 'LINK=#W03'
DEFINE WORK FILE 3 'FILE=#W03'
```

Link-Name	<p>Beispiel:</p> <pre>DEFINE WORK FILE 1 'LINKW01'</pre> <p>bedeutet dasselbe wie</p> <pre>DEFINE WORK FILE 1 'LINK=LINKW01'</pre> <p>Eine Datei mit dem Link LINKW01 muss zur Laufzeit vorhanden sein. Dieser kann entweder mittels einer JCL vor dem Start von Natural oder durch eine dynamische Zuweisung von der aktuellen Anwendung erstellt werden. Für eine dynamische Zuweisung kann die Programmierschnittstelle (API) USR2021N (dynamische Dateizuweisung) in der Library SYSEXT verwendet werden. Wenn der Link zu einer anderen Datei, z.B. W01, vor der Ausführung aktiv war, wird er in Abhängigkeit vom Wert des Subparameters FREE des Profilparameters WORK (mögliche Werte sind ON und OFF) entweder freigegeben oder beibehalten. Die Freigabe erfolgt über einen expliziten RELEASE-Aufruf an den BS2000/OSD-Kommandoprozessor.</p>
------------------	--

Dateiname	<p>Beispiel:</p> <pre>DEFINE WORK FILE 2 'NATURALvr.TEST.WORK02'</pre> <p>bedeutet dasselbe wie</p> <pre>DEFINE WORK FILE 2 'FILE=NATURALvr.TEST.WORK02'</pre> <p>dabei steht <i>vr</i> für die Natural-Versions- und Release-Nummer.</p> <p>Die in <i>operand1</i> angegebene Datei wird mittels eines FILE-Makroaufrufs aufgebaut und erbt den Link-Namen, der für die entsprechende Arbeitsdatei vor Ausführung des DEFINE WORK FILE-Statements gültig war.</p>
Generischer Dateiname	<p>Beispiel:</p> <pre>DEFINE WORK FILE 21 '*'</pre> <p>bedeutet dasselbe wie</p> <pre>DEFINE WORK FILE 21 'FILE=*'</pre> <p>Eine Datei mit einem nach Regel 4 erstellten Namen wird mittels eines FILE-Makroaufrufs aufgebaut und erbt den Link-Namen, der für die entsprechende Arbeitsdatei vor Ausführung des DEFINE WORK FILE-Statements gültig war.</p> <pre>DEFINE WORK FILE 22 'FILE=*,LINK=WFLK22'</pre> <p>Eine Datei mit einem nach Regel 4 erstellten Namen wird mittels eines FILE-Makroaufrufs mit dem angegebenen Link-Namen aufgebaut.</p>
Dateiname und Link-Name	<p>Beispiel:</p> <pre>DEFINE WORK FILE 11 'NATURALvr.TEST.WORKF11,LNKW11'</pre> <p>bedeutet dasselbe wie</p> <pre>DEFINE WORK FILE 11 'FILE=NATURALvr.TEST.WORKF11,LINK=LNKW11'</pre> <p>bedeutet dasselbe wie</p> <pre>DEFINE WORK FILE 11 'FILE=NATURALvr.TEST.WORKF11,LNKW11'</pre> <p>dabei steht <i>vr</i> für die Natural-Versions- und Release-Nummer.</p> <p>Die in <i>operand1</i> angegebene Datei wird mittels eines FILE-Makroaufrufs mit angegebenem Link-Namen aufgebaut und der entsprechenden Arbeitsdateinummer zugewiesen.</p>

Generischer Dateiname und Link-Name	<p>Beispiel:</p> <pre>DEFINE WORK FILE 27 '*,*'</pre> <p>bedeutet dasselbe wie</p> <pre>DEFINE WORK FILE 27 'FILE=*,LINK=*'</pre> <p>Eine Datei mit nach Regel 4 und Regel 5 erstelltem Dateinamen und Link-Namen wird mittels eines FILE-Makroaufrufs aufgebaut und der angegebenen Arbeitsdatei 27 zugewiesen.</p> <p>Anmerkung: Wenn Dateiname und Link-Name angegeben werden, wird der vorherige Link-Name nicht freigegeben, ungeachtet des Wertes des Subparameters FREE des Profilparameters WORK.</p>
*DUMMY	Um ein Dummy-Dataset anzugeben.

Arbeitsdateiname unter CICS

Für eine mit der Zugriffsmethode AM=CICS definierte Arbeitsdateinummer kann *operand1* in Abhängigkeit vom Subparameter TYPE des Profilparameters WORK für die Arbeitsdatei ein Übergangsdaten- oder Zwischenspeicher-Warteschlangen-Name (1 bis 8 Zeichen) sein. Für TYPE=TD (Übergangsdaten) werden nur die ersten 4 Zeichen von *operand1* berücksichtigt und das Ausgabemedium für die Übergangsdaten muss vorher für CICS definiert worden sein.

Weitere Informationen über Arbeitsdateien siehe auch den Abschnitt *Natural Print and Work Files under CICS* in der *Natural CICS Interface*-Dokumentation.

Arbeitsdateiname unter Com-plete/SMARTS

Unter Com-plete stehen PFS-Dateien mit der Zugriffsmethode AM=SMARTS zur Verfügung. Es kann ein Arbeitsdateiname zugewiesen werden, auch wenn er für Natural nicht definiert worden ist.

```
DEFINE WORK (14) '/nat/path/workfile'
DEFINE WORK (14) 'workfile'
```

Es ist vom Parameter MOUNT_FS von SMARTS abhängig, ob die Datei auf einem SMARTS Portable File System oder auf dem Native File System residiert. Das erste Element des Pfads (/nat/) legt das Zielfeld-System fest.

Wenn die Zeichenkette nicht mit einem Schrägstrich (/) beginnt, wird der Pfad der Datei von der Umgebungsvariablen \$NAT_WORK_ROOT genommen.

Der angegebene Pfadname muss vorhanden sein. Wenn das Statement DEFINE WORK FILE ausgeführt wird, wird die Datei dynamisch zugewiesen. Wenn das angegebene Member nicht vorhanden ist, wird ein neues Member dieses Namens angelegt.