Deutsch

FUJITSU Software

BS2000 OSD/BC V10.0

DVS-Makros

Benutzerhandbuch

Ausgabe Juni 2016

Kritik… Anregungen… Korrekturen…

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an manuals@ts.fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2008

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2008 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright und Handelsmarken

Inhalt

1 Einleitung

Das vorliegende Handbuch beschreibt die Makroaufrufe des Datenverwaltungssystems des BS2000.

Das Datenverwaltungssystem (DVS) ist ein selbstständiges Teilsystem im BS2000. Es ist das Bindeglied zwischen den vom Benutzer gesteuerten Zugriffen auf Datenobjekte und den zentralen Gerätetreibern des Basissystems. Auch das DVS selbst nutzt Dienste des Basissystems.

Das Datenverwaltungssystem als „verteiltes System“

Das DVS unterstützt in besonderem Maße die Datenverarbeitung auf gemeinschaftlichen Datenträgern, den „Public Volume Sets“ (kurz: Pubsets). In MPVS-Systemen (Multiple Public Volume Sets) sind mehrere solcher Pubsets zu einem System zusammengefasst.

MPVS-Systeme zeichnen sich dadurch aus, dass die einzelnen Pubsets unabhängig voneinander verwaltet werden können. Lediglich der sog. Home-Pubset und die Pubsets, die Systemdaten enthalten, müssen während des gesamten Systemlaufs zur Verfügung stehen. Das System kann auf verschiedene Pubsets verteilt sein. Die Pubsets, die keine Systembereiche enthalten, kann der Systemverwalter nach Bedarf hinzuschalten oder wieder aus dem System entfernen. Durch die Verteilung des Systems und die Unabhängigkeit der Pubsets voneinander ist eine hohe Ausfallsicherheit gewährleistet.

Für den Benutzer des BS2000 bedeutet das: Solange er nicht explizit private Datenträger anfordert, werden seine Dateien auf dem Pubset angelegt, der ihm vom Systemverwalter als Standard-Pubset zugewiesen wurde. Er muss weder Datenträger noch Geräte anfordern. Auch die Speicherplatzverwaltung übernimmt das DVS.

In jedem Pubset wird für jeden Benutzer, der auf diesen Pubset zugreifen darf, ein Dateikatalog geführt. Durch Benutzerkennung und Katalogkennung – das ist die Kennung des Pubsets, auf dem der Katalog geführt wird – sind alle Benutzerdateien eindeutig gekennzeichnet. Gleichzeitig ist gewährleistet, dass nur dem Dateieigentümer Dateizugriff gestattet ist, sofern er nicht explizit die Datei für „Fremdzugriffe“ freigibt.

Funktionen des Datenverwaltungssystems

Das Datenverwaltungssystem ermöglicht dem Benutzer die Verarbeitung seiner Daten durch die Funktionen, die für die Dateiverarbeitung bereitgestellt werden oder nötig sind:

– Erzeugen und Verwalten von Dateien inkl. Speicherplatzverwaltung
– Verwalten von Katalogen
– Bereitstellen von Dateien und Dateibearbeitung über die Zugriffsmethoden
– Zuordnen von Dateien zu Programmen

Darüber hinaus bietet das DVS dem Benutzer die Möglichkeit, Datenschutz und Dateischutz-Merkmale auf Dateiebene zu definieren. Die Datensicherheit wird vom DVS z.B. beim Dateizugriff durch das Setzen von Sperren unterstützt.

Die Funktionen des DVS sind realisiert über die hier beschriebenen Programmschnittstellen (Assembler) und über funktionsgleiche Kommandoschnittstellen, die in den Handbüchern „Kommandos“ [3] beschrieben sind.

1.1 Zielsetzung und Zielgruppen und des Handbuchs

Das Handbuch wendet sich an den Benutzer, der mithilfe der Assembler-Programmierschnittstelle Dateien verarbeiten oder verwalten und Datenschutz, Dateischutz, Datensicherheit usw. über DVS-Makros steuern will.

1.2 Konzept des Handbuchs

Kapitel 2: Übersicht über die DVS-Makroaufrufe

Dieser Teil des Handbuches enthält eine alphabetische und funktionale Zusammenstellung aller DVS-Makros.

Kapitel 3: Hinweise zur Programmierung

In diesem Kapitel werden die einzelnen Zugriffsmethoden aufgeführt, mit denen das DVS arbeitet. Hier sind die Makroaufrufe und Operanden aufgeführt, die bei der jeweiligen Zugriffsmethode möglich sind. Hauptsächlich sind hier programmierrelevante Besonderheiten der einzelnen Zugriffsmethoden dargestellt.

Kapitel 4: Makroaufrufe

Dieser Teil des Handbuches ist als Nachschlageteil konzipiert. Er enthält Funktionsbeschreibung, Syntax und Operandenbeschreibung der DVS-Makros. Sämtliche Makroaufrufe sind in alphabetischer Reihenfolge aufgeführt.

Der Anhang enthält Tabellen und Listen, auf die an mehreren Stellen des Handbuches Bezug genommen wird. Dazu gehören insbesondere Metasyntax sowie Tabellen mit Kennsatzformaten für Banddateien und DVS-Fehlermeldungen

Den Abschluss des Handbuchs bilden Fachwörter-, Literatur- und Stichwortverzeichnis.

Hinweise zur Benutzung des Handbuchs

Literaturhinweise sind im Text in der Regel als Kurztitel mit einer Referenznummer angegeben. Das Literaturverzeichnis enthält die vollständigen Titel.

Eine allgemeine Beschreibung der Funktionen des DVS ist im Handbuch „Einführung in das DVS“ [1] zu finden.

Readme-Datei

Funktionelle Änderungen der aktuellen Produktversion und Nachträge zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei.

Readme-Dateien stehen Ihnen online bei dem jeweiligen Produkt zusätzlich zu den Produkthandbüchern unter *http://manuals.ts.fujitsu.com* zur Verfügung. Alternativ finden Sie Readme-Dateien auch auf der Softbook-DVD.

Informationen unter BS2000

Wenn für eine Produktversion eine Readme-Datei existiert, finden Sie im BS2000-System die folgende Datei:

`SYSRME.<product>.<version>.<lang>`

Diese Datei enthält eine kurze Information zur Readme-Datei in deutscher oder englischer Sprache (<lang>=D/E). Die Information können Sie am Bildschirm mit dem Kommando `SHOW-FILE` oder mit einem Editor ansehen.
Das Kommando `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` zeigt, unter welcher Benutzerkennung die Dateien des Produkts abgelegt sind.

Ergänzende Produkt-Informationen

Aktuelle Informationen, Versions-, Hardware-Abhängigkeiten und Hinweise für Installation und Einsatz einer Produktversion enthält die zugehörige Freigabemitteilung. Solche Freigabemitteilungen finden Sie online unter *http://manuals.ts.fujitsu.com*.

1.3 Änderungen gegenüber dem Vorgänger-Handbuch

Das Handbuch „DVS-Makros“ wurde zuletzt zu BS2000/OSD-BC V9.0 aufgelegt. Folgende wesentliche Änderungen haben sich seit der letzten Ausgabe ergeben:

Änderungen an den Makro-Schnittstellen

Makro / Operand oder RC	Inhalt der Änderung
IMPNFIL	Neuer Makro: Importiert Node-Files von einem Net-Storage-Volume
FILE	Neuer Operand NFTYPE bestimmt den Dateityp einer Datei auf Net-Storage. Die Datei kann als BS2000-Datei oder als Node-File angelegt werden.
FSTAT	Neuer Operand FILTYPE zur Auswahl von BS2000-Dateien oder Node-Files.
ERASE	Neuer Operand FILTYPE zur Auswahl von BS2000-Dateien oder Node-Files. Bei DELETE-OR-EXPORT bleiben Node-Files erhalten (entspricht der Funktion des Kommandos EXPORT-NODE-FILES).
COPFILE	Neuer Operand CHDATE gibt an, ob die Zieldatei das Änderungsdatum der Quelldatei erhält. Die Angabe PROTECT=*SAME-AND-CHANGE-DATE wird noch kompatibel unterstützt.

Grundlegende Informationen zum Einsatz von Net-Storage in BS2000 finden Sie im Handbuch „Einführung in die Systembetreuung“ [7]. Das Arbeiten mit BS2000-Dateien und Node-Files auf Net-Storage ist im Handbuch „Einführung in das DVS“ [1] beschrieben.

Die Geräte- und Volumetyp-Tabelle finden Sie im Handbuch „Systeminstallation“ [16]. Informationen zu den Volumetypen des DMS (Net-Storage und Bandverarbeitung) finden Sie im Handbuch „Kommandos“ [3].

Allgemeine Änderung

Die bisherige Bezeichnung BS2000/OSD-BC des BS2000-Grundausbaus ändert sich und lautet ab Version V10.0: BS2000 OSD/BC.

Vorgängerversionen werden mit der bisherigen Bezeichnung BS2000/OSD-BC zitiert.

1.4 Darstellungsmittel

In diesem Handbuch werden folgende Darstellungsmittel verwendet:

Dieses Zeichen kennzeichnet Hinweise auf wichtige Informationen

Dieses Zeichen kennzeichnet einen Warnhinweis, der auf die Möglichkeit des Datenverlustes oder anderer ernsthafter Schäden an Daten hinweist.

[ ] Literaturhinweise werden im Text in Kurztiteln angegeben. Der vollständige Titel jeder Druckschrift, auf die durch eine Nummer verwiesen wird, ist im Literaturverzeichnis hinter der entsprechenden Nummer aufgeführt.

`Eingabe` In Anwendungsbeispielen sind Eingaben an das System und Ausgaben des Systems in `Schreibmaschinenschrift` dargestellt.

Informationen zur Syntaxdarstellung der Makroaufrufe finden Sie im Anhang (ab Seite 862).

2 Übersicht über die DVS-Makros

2.1 DVS-Makros alphabetisch geordnet

Makroaufruf	Kurzbeschreibung
ADDPLNK	*ISAM:* Poolkettungsnamen definieren
BTAM	Alle BTAM-Aktionen steuern
CATAL	Katalogeintrag bearbeiten
CHKFAR	Zugriffsrechte auf Datei überprüfen
CHNGE	TFT-Eintrag ändern
CLOSE	Datei schließen
COMPFIL	Zwei Plattendateien vergleichen
COPFILE	Datei kopieren
CREAIX	*ISAM:* Sekundärschlüssel für ISAM-Datei erzeugen
CREPOOL	*ISAM:* ISAM-Pool erzeugen
DECFILE	Verschlüsselte Datei in unverschlüsselte Datei umwandeln
DELAIX	*ISAM:* Sekundärschlüssel einer ISAM-Datei löschen
DELPOOL	*ISAM:* ISAM-Pool löschen/freigeben
DIV	Dateizugriff über virtuellen Adressraum
DROPTFT	TFT-Eintrag freigeben
EAM	Makroaufruf (Typ R)
ELIM	*ISAM:* Satz streichen
ENCFILE	Unverschlüsselte Datei in verschlüsselte Datei umwandeln
ERASE	Dateien löschen
EXLST	Exit-Adressenliste anlegen (Typ O)
EXRTN	Rücksprung aus Fehlerroutinen (Typ R)
FCB	Dateisteuerblock definieren (Typ O)
FCBAD	FCB-Adressen anlegen (Typ O)

(Teil 1 von 3)

Makroaufruf	Kurzbeschreibung
FEOV	*BTAM/SAM:* Band abschließen
FILE	Dateimerkmale definieren / Dateiverarbeitung steuern
FILELST	Variable Operandenbereiche für FILE-Makro erzeugen
FPAMACC	*FASTPAM:* Dateizugriffe formulieren
FPAMSRV	*FASTPAM:* Verwaltungsaufrufe formulieren
FSTAT	Kataloginformation anfordern
GET	*ISAM/SAM:* Nächsten Satz lesen
GETFL	*ISAM:* Lesen Satz nach Markierung
GETKY	*ISAM:* Satz lesen mit angegebenem Schlüssel
GETR	*ISAM:* Sequenziell „rückwärts“ Lesen
IDBPL	*BTAM:* BTAM-Operandenliste (Typ O)
IDFCB	Versorgung eines FCB mit symbolischen Namen (Typ O)
IDFCBE	Versorgung der FCBE mit symbolischen Namen (Typ O)
IDMCB	Versorgung des MFCB (EAM-Steuerblock mit symbolischen Namen)
IDPPL	*UPAM:* PAM-Operandenliste
IMPNFIL	Katalogeintrag für Node-Files erstellen (importieren)
IMPORT	Katalogeintrag für Dateien erstellen (importieren)
INSRT	*ISAM:* Satz einfügen (Typ R)
ISREQ	*ISAM:* Sperre aufheben (Typ O)
LBRET	Rückkehr aus Benutzer-Kennsatzroutine (Typ R)
LFFSNAP	Dateien von einem Snapset auflisten
LJFSNAP	Jobvariablen von einem Snapset auflisten
MAILFIL	Datei per E-Mail an eine Benutzerkennung versenden
NDWERINF	*BTAM:* Status-Bytes abfragen
OPEN	Datei eröffnen (Typ R)
OSTAT	*ISAM:* Information über eröffnete Dateien (Typ R)
PAM	*UPAM:* UPAM-Aktionen ausführen
PUT	*ISAM/SAM:* Satz schreiben
PUTX	*ISAM/SAM:* Satz ersetzen
RDTFT	Informationen aus TFT und TST
RELTFT	TFT-Eintrag löschen
REMPLNK	*ISAM:* Poolkettungsname löschen

(Teil 2 von 3)

Makroaufruf	**Kurzbeschreibung**
RETRY	*ISAM:* Makroaufruf wiederholen
RELSE	Block abschließen
RFFSNAP	Dateien von einem Snapset restaurieren
RJFSNAP	Jobvariablen von einem Snapset restaurieren
SETL	*ISAM/SAM:* Positionieren in der Datei
SHOPLNK	Informationen über ISAM-Pool-Kettungsnamen ausgeben
SHOPOOL	Informationen über ISAM-Pool ausgeben
SHOWAIX	Informationen über Sekundärschlüssel ausgeben
STORE	*ISAM:* Satz speichern
VERIF	Datei wiederherstellen

(Teil 3 von 3)

2.2 DVS-Makros funktional geordnet

Die folgenden Tabellen bieten eine Übersicht über die Funktionen der in diesem Handbuch beschriebenen Makros.

Nähere Informationen sind den Makro- und Operandenbeschreibungen in diesem Handbuch zu entnehmen oder den entsprechenden Einführungskapiteln im Handbuch „Einführung in das DVS“ [1].

2.2.1 Wartung des Dateibestandes

Zur Wartung des Dateibestandes gehören nicht nur Erstellen, Kopieren, Löschen oder Restaurieren von Dateien, sondern auch die Pflege des Dateikatalogs durch den Benutzer.

Makro	Operanden	Kurzbeschreibung
CATAL		erstellt oder ändert Katalogeinträge
COMPFIL		vergleicht zwei Plattendateien
COPFILE		kopiert Dateien
ERASE		löscht Dateien / exportiert Dateien
FILE		erstellt einen Katalogeintrag und reserviert Speicherplatz für nicht katalogisierte Dateien
	SPACE	reserviert Speicherplatz oder gibt ihn frei
	STATE	erstellt einen Katalogeintrag für Dateien auf privaten Datenträgern
FSTAT		gibt Informationen aus dem Dateikatalog aus
IMPORT		erstellt Katalogeinträge für Dateien (importieren)
LFFSNAP LJFSNAP		listet Dateien von einem Snapset listet Jobvariablen von einem Snapset
MAILFIL		sendet Datei per E-Mail an eine Benutzerkennung
RFFSNAP RJFSNAP		restauriert Dateien von einem Snapset restauriert Jobvariablen von einem Snapset
VERIF		rekonstruiert beschädigte Dateien

2.2.2 Steuerung der Dateiverarbeitung

Für die Verarbeitung von Dateien durch ein Programm muss eine Verbindung zwischen Datei und Programm hergestellt werden können. Diese Verbindung kann im FCB festgelegt oder über FILE-Makro hergestellt werden, wenn das Programm mit einem programminternen Dateinamen (= Dateikettungsname) arbeitet. Die Verbindung wird neben anderen Informationen in der Task File Table (TFT) hinterlegt. Die Steuerung der Dateiverarbeitung erfolgt also über die TFT.

Die Steuerung der Dateiverarbeitung umfasst für NK-ISAM-Dateien außerdem die Verwaltung von Benutzer-ISAM-Pools, in denen diese Dateien verarbeitet werden. Der Benutzer kann auch Standard-ISAM-Pools des Systems nutzen, hat dann jedoch keinen Einfluss auf Poolgröße und -belegung.

Makro	**Operanden**	**Kurzbeschreibung**
ADDPLNK		weist einem Benutzer-ISAM-Pool einen Poolkettungsnamen zu
CHNGE		weist einer Datei einen neuen Dateikettungsnamen zu
CREAIX		erzeugt einen Sekundärindex für eine ISAM-Datei
CREPOOL		richtet einen Benutzer-ISAM-Pool ein
DELAIX		löscht Sekundärindizes einer ISAM-Datei
DELPOOL		löscht einen Benutzer-ISAM-Pool
FCB	FILE	feste Zuordnung von Datei und Programm
	LINK	definiert den Dateikettungsnamen im Programm
	POOLLNK	stellt die Verbindung zum Benutzer-ISAM-Pool her
FILE	LINK	erstellt einen TFT-Eintrag, weitere Operanden beschreiben Datei- und Verarbeitungseigenschaften
	BLKCTRL	legt das Datenformat fest
	NFTYPE	legt den Dateityp auf Net-Storage fest (BS2000 oder Node-File)
	POOLLNK	stellt die Verbindung zum Benutzer-ISAM-Pool her
RDTFT		gibt TFT-Informationen aus
RELTFT		löscht einen TFT-Eintrag
REMPLNK		löscht den Poolkettungsnamen
SHOPLNK		informiert über die Zuordnung von ISAM-Pools zu Poolkettungsnamen
SHOPOOL		informiert über Attribute und Belegungszustände von ISAM-Pools

2.2.3 Unterstützung von Datenschutz und Datensicherheit

Die vom DVS automatisch unterstützten Mechanismen für Datei- und Datenschutz (Überprüfung der Zugriffsberechtigung usw.) kann der Anwender erweitern, z.B. durch die Vergabe von Kennwörtern. Datensicherheit wird dem Anwender durch verschiedene Mechanismen zur Wiederherstellung von Dateien oder Programmen usw. gewährleistet.

Dateischutz

Makro	Operanden	Kurzbeschreibung
CATAL	SHARE	gibt Dateizugriff für fremde Benutzerkennungen frei
	ACCESS	bestimmt die zulässige Zugriffsart (Lesen/Schreiben)
	OWNERAR GROUPAR OTHERAR	} legt in der BASIC-ACL Zugriffsrechte für die Benutzergruppen fest (siehe Handbuch „Einführung in das DVS“ [1])
	GUARDS	Bei Einsatz von SECOS: Erweiterter Zugriffsschutz für Dateien
	EXPASS RDPASS WRPASS	} Kennwörter für die verschiedenen Zugriffsebenen
	RETPD	legt eine Schreibschutzfrist fest
	PROTECT	Übernahme von Schutzattributen
CHKFAR		überprüft die Zugriffsrechte des Aufrufers auf Dateien
COPFILE	PROTECT	übernimmt beim Kopieren die Schutzmerkmale der Originaldatei
DECFILE		wandelt verschlüsselte Datei in unverschlüsselte Datei um
ENCFILE		wandelt unverschlüsselte Datei in verschlüsselte Datei um
FILE	RETPD	legt eine Schreibschutzfrist fest (nur in Zusammenhang mit Dateieröffnung!)
FCB	PASS	ermöglicht Zugriff bei Kennwortschutz
	RETPD	legt eine Schreibschutzfrist fest

Datenschutz

Makro	Operanden	Kurzbeschreibung
CATAL	DESTROY	legt im Katalogeintrag fest, dass Plattendateien beim Löschen überschrieben werden bzw. auf Bändern fremde Restdaten überschrieben werden (siehe FILE: DESTOC)
ERASE	DESTROY	Datenzerstörung beim Löschen
DECFILE		wandelt verschlüsselte Datei in unverschlüsselte Datei um
ENCFILE		wandelt unverschlüsselte Datei in verschlüsselte Datei um
FILE	DESTOC	beim Bandwechsel oder nach dem Schließen der Datei werden auf dem Band folgende (alte) Daten überschrieben

Datensicherheit

Makro	Operanden	Kurzbeschreibung
CATAL	BACKUP	definiert die Häufigkeit der automatischen Sicherung
COPFILE	REPLACE	legt fest, ob eine existente Datei beim Kopieren überschrieben wird
CREPOOL	WROUT	sofortiges Rückschreiben bei Aktualisierung von ISAM-Dateien
EXLST		definiert Exit-Routinen für Fehler und andere Ereignisse
FCB	EXIT	Adresse einer Exit-Routine oder eines EXLST-Makroaufrufs
	WRCHK	Kontrollesen, um Aufzeichnungsfehler beim Schreiben auf Platte zu vermeiden
FILE	WRCHK	Kontrollesen, um Aufzeichnungsfehler beim Schreiben auf Platte zu vermeiden
	WROUT	sofortiges Rückschreiben bei Aktualisierung von ISAM-Dateien
VERIF		rekonstruiert Dateistrukturen / entsperrt Dateien
WRCPT		schreibt einen Fixpunkt / erstellt eine Fixpunktdatei für den Wiederanlauf mit dem Kommando RESTART-PROGRAM

2.2.4 Geräte- und Datenträgerverwaltung

Das DVS unterstützt den Benutzer bei der Verarbeitung von Dateien auf privaten Datenträgern durch die Möglichkeit, Geräte und Datenträger für seinen Auftrag zu reservieren.

Makro	Operanden	Kurzbeschreibung
FILE	DEVICE VOLUME	bestimmt Gerätetyp und Datenträger für eine Datei auf privaten Datenträgern oder auf Net-Storage-Volumes
	MOUNT	fordert die Bereitstellung privater Datenträger an der Konsole
IMPNFIL		erstellt Katalogeintrag für Node-Files (importieren)
IMPORT		erstellt Katalogeinträge für Dateien (importieren)
RELTFT		löscht TFT-Einträge und gibt implizit Geräte frei

2.2.5 Zugriff zu Dateien

Der Dateizugriff erfolgt durch die Aktionsmakroaufrufe der verschiedenen Zugriffsmethoden. Das DVS übernimmt aber auch das Öffnen und Schließen einer Datei (OPEN/CLOSE-Behandlung) in Abhängigkeit von der jeweiligen Zugriffsart.

DVS-Makros der Dateiverarbeitung („Service-Makros“)

Die DVS-Makros der Dateiverarbeitung sind für mehrere Zugriffsmethoden gültige Makroaufrufe.

Makro	Kurzbeschreibung
CLOSE	schließt eine oder mehrere Dateien
EXLST	definiert Fehlerausgänge
EXRTN	Rücksprung aus EXLST-Routinen
FCB	legt einen Dateisteuerblock an (File Control Block = FCB)
FCBAD	legt den FCB im Literalbereich eines Programms an
LBRET	Rücksprung aus Benutzer-Kennsatzroutinen (Bandverarbeitung)
OPEN	eröffnet eine Datei

Zugriffsmethoden-spezifische Makroaufrufe

Es werden folgende Zugriffsmethoden unterschieden:

– BTAM
– DIV
– EAM
– FASTPAM
– SAM
– ISAM
– UPAM

Für verschlüsselte Dateien erfolgt der Zugriff in gewohnter Form (mit Klartext-Inhalt). Das Entschlüsseln beim Lesen und das Verschlüsseln beim Schreiben werden intern und automatisch ausgeführt. Die Makros zum Dateizugriff in einem Programm brauchen dafür nicht geändert zu werden. Vor dem Öffnen der Datei muss nur das zugehörige Crypto-Kennwort in die Crypto-Kennworttabelle der zugreifenden Task eingetragen werden. Dies kann vor dem Starten des Programms erfolgen.

In den folgenden Tabellen sind die Makroaufrufe für den Dateizugriff den einzelnen Zugriffsmethoden zugeordnet.

BTAM = Basic Tape Access Method

BTAM ist eine Zugriffsmethode für blockorientierte Bandverarbeitung; mit BTAM lassen sich auch Banddateien verarbeiten, die nicht mit BTAM erstellt wurden. Während der Verarbeitung einer Banddatei kann die Verarbeitungsrichtung innerhalb der Datei beliebig gewechselt werden. Die Bänder können beliebig block- oder abschnittsweise positioniert werden. BTAM verarbeitet Dateien mit oder ohne Standardblockung.

Makro	**Kurzbeschreibung**
BTAM	steuert alle BTAM-Aktionen
FEOV	Bandwechsel auslösen
NDWERINF	Status-Bytes abfragen

DIV = Data In Virtual

Grundlage der Dateiverarbeitung für einen Anwender sind 4KByte große Blöcke. Mit DIV lassen sich auch Dateien verarbeiten, die nicht mit DIV erstellt wurden.

Makro	**Kurzbeschreibung**
DIV	Dateien mit der Zugriffsmethode DIV bearbeiten – Datei eröffnen – Fenster (Arbeitsbereich im virtuellen Adressraum) definieren – Modifizierte Seiten aus dem Fenster in die Datei auf Platte zurückschreiben – Änderungen im Fenster zurücknehmen – Fenster im virtuellen Adressraum freigeben – Datei schließen; ggf. noch existierende Fenster werden mit Standardwerten freigegeben.

EAM = Evanescent Access Method

Mit EAM werden taskspezifische temporäre Dateien im SYSEAM-Bereich verarbeitet. EAM ist eine blockorientierte Zugriffsmethode und eignet sich vor allem für die schnelle Verarbeitung von tasklokalen Arbeitsdateien.

Makro	**Kurzbeschreibung**
EAM	steuert alle EAM-Zugriffe

FASTPAM = Fast Primary Access Method

FASTPAM ist eine blockorientierte Zugriffsmethode. Es arbeitet immer mit 4KB-Blöcken. Mit FASTPAM lassen sich auch Dateien verarbeiten, die nicht mit FASTPAM erstellt wurden.

Makro	**Kurzbeschreibung**
FPAMACC	Dateizugriffe formulieren – synchrones Lesen und Schreiben von logischen Blöcken – asynchrones Lesen und Schreiben von logischen Blöcken – Warten auf die Beendigung asynchroner Ein-/Ausgabeaufträge – Benachrichtigung über die Beendigung asynchroner Ein-/Ausgabeaufträge
FPAMSRV	Verwaltungsaufrufe formulieren – Systemumgebung (FASTPAM-Environment) vorbereiten – Ein-/Ausgabebereiche (FASTPAM-IO-Area-Pool) vorbereiten – Datei zur Bearbeitung eröffnen – eine mit FPAMSRV geöffnete Datei schließen – Systemumgebung (FASTPAM-Environment) abbauen – Ein-/Ausgabebereiche (FASTPAM-IO-Area-Pool) abbauen

SAM = Sequential Access Method

SAM ist eine satzorientierte Zugriffsmethode. Die Sätze sind sequenziell in der Datei abgelegt. SAM ermöglicht dem Benutzer die Verarbeitung aufeinander folgender Datensätze, und zwar sowohl in Richtung Dateiende als auch Dateianfang. SAM erfüllt für die Bandverarbeitung alle von DIN 66029 bis Austauschstufe 3 gestellten Anforderungen; es können sowohl Dateien mit Standardblöcken als auch mit Nichtstandardblöcken verarbeitet werden.

Makro	**Kurzbeschreibung**
FEOV	Bandwechsel auslösen
GET	stellt den nächsten Satz bereit
PUT	schreibt den nächsten Satz
PUTX	(nur im Locate-Mode) ersetzt einen durch GET gelesenen Satz
RELSE	schließt einen Datenblock ab
SETL	positioniert auf Dateianfang, -ende oder auf einen Satz

ISAM = Indexed Sequential Access Method

Grundlage der Verarbeitung ist eine aus Index- und Datenblöcken aufgebaute Datei. Jeder Datensatz enthält einen Schlüssel; die Schlüssel dienen als Sortierkriterium (zu Index- und Datenblock siehe Handbuch „Einführung in das DVS“ [1]).

Makro	**Kurzbeschreibung**
ADDPLNK	weist einem Benutzer-ISAM-Pool einen Poolkettungsnamen zu
CREAIX	erzeugt einen Sekundärindex für eine ISAM-Datei
CREPOOL	richtet einen Benutzer-ISAM-Pool ein
DELAIX	löscht Sekundärindizes einer ISAM-Datei
DELPOOL	löscht einen Benutzer-ISAM-Pool
ELIM	streicht einen Satz aus der Datei
GET	liest sequenziell die Sätze der Datei.
GETFL	bei Verwendung von markierten ISAM-Schlüsseln: liest den folgenden Satz innerhalb des Markierungsbereichs (sequenziell)
GETKY	liest den ersten Satz mit dem angegebenen Schlüssel
GETR	liest sequenziell die Sätze rückwärts
INSRT	fügt einen Satz mit neuem ISAM-Schlüssel in die Datei ein
ISREQ	hebt eine ISAM-Sperre auf
OSTAT	informiert über Anzahl und Art gleichzeitiger Dateizugriffe
PUT	schreibt sequenziell Sätze an das Dateiende (inklusive Prüfung der Schlüssel auf gültige Reihenfolge)
PUTX	ersetzt einen zuvor gelesenen Satz
REMPLNK	löscht den Poolkettungsnamen
RETRY	setzt nach Durchlaufen des EXLST-PGLOCK-Ausgangs den ISAM-Zeiger neu und wiederholt den letzten Makroaufruf
SETL	positioniert den ISAM-Zeiger für die weitere Verarbeitung auf Dateianfang, -ende oder einen bestimmten Satz
SHOPLNK	informiert über die Zuordnung von ISAM-Pools zu Poolkettungsnamen
SHOPOOL	informiert über Attribute und Belegungszustände von ISAM-Pools
SHOWAIX	informiert über Sekundärindizes einer ISAM-Datei
STORE	– fügt einen Satz mit neuem ISAM-Schlüssel in die Datei ein – überschreibt einen Satz mit bereits vorhandenem ISAM-Schlüssel, wenn Mehrfachvergabe von ISAM-Schlüsseln nicht zulässig ist – fügt einen Satz mit bereits vorhandenem ISAM-Schlüssel als letzten Satz mit diesem Schlüssel in die Datei ein

UPAM = User Primary Access Method

UPAM ist eine blockorientierte Zugriffsmethode. Grundlage von UPAM ist der Standardblock (= PAM-Seite). Mit UPAM lassen sich auch Dateien verarbeiten, die nicht mit UPAM erstellt wurden.

Makro	**Kurzbeschreibung**
PAM	steuert alle UPAM-Zugriffe

Bei ereignisgesteuerter Verarbeitung sind zusätzlich folgende Makroaufrufe von Bedeutung (ausführliche Beschreibung siehe Handbuch „Makroaufrufe an den Ablaufteil“ [2]):

Makro	**Kurzbeschreibung**
CHKEI	prüft die Warteschlangenbelegung einer Ereigniskennung
CONTXT	greift auf den Registersatz des unterbrochenen Prozesses zu
DISCO	schließt die Routine für den Contingency-Prozess
DISEI	trennt des Benutzerprogramm von der Ereigniskennung
ENACO	eröffnet eine Routine als Contingency-Prozess und weist ihr Namen und Priorität zu
ENAEI	richtet Ereigniskennung ein und/oder stellt Verbindung her zwischen aufrufendem Prozess und Ereigniskennung
FECB	Steuerblock für die Ereigniskennung (= File Event Control Block)
LEVCO	ändert die Priorität des aufgerufenen Prozesses
POSSIG	signalisiert ein Ereignis
RETCO	beendet aufrufenden Contingency-Prozess
SOLSIG	fordert Signal von der Ereigniskennung an
SUSPEND	versetzt den aufrufenden Prozess in einen unterbrechbaren Wartezustand

2.2.6 Generierung der Operandenlisten für Steuerblöcke, DVS-Tabellen usw.

Der Anwender kann Programmbereiche oder DUMMY-Sections (DSECTs) erzeugen, die es ihm ermöglichen auf Inhalte von DVS-Tabellen, Dateisteuerblöcken usw. oder auf die Operandenlisten von DVS-Makros mit symbolischen Adressen zuzugreifen.
Für die meisten Makroaufrufe ist dies möglich über den MF-Operanden, anderenfalls gibt es spezielle DSECT-Makroaufrufe. Bei „älteren“ Makroaufrufen (z.B. CATAL), die erst in einer späteren Version umgestellt wurden, entscheidet der VERSION-Operand, ob der spezielle DSECT-Makroaufruf weiter genutzt werden muss oder ob die Angabe über den MF-Operanden möglich ist.

Operandenlisten für DVS-Makros mit Kommandofunktion

Makroaufruf	**erforderliche VERSION-Angabe**	**MF-Operand**	**„DSECT-Makro“**
ADDPLNK		x	-
CATAL	ohne Operand VERSION	-	IDCAT
	VERSION≥1	x	-
CHNGE		-	IDCHA
COMPFIL		x	-
COPFILE		x	-
COPY	*(Makro ersetzt durch COPFILE)*	-	IDCOP
CREAIX		x	-
CREPOOL		x	-
DECFILE		x	-
DELPOOL		x	-
DELAIX		x	-
DROPTFT		x	-
ENCFILE		x	-
ERASE	VERSION=0	-	IDERS
	VERSION≥1	x	-
FILE	VERSION=0	-	IDPFL/IDPFX
	VERSION≥1	x	-
FSTAT	VERSION=0 (entspricht 710)	-	IDFST
	VERSION≥1 (1 entspricht 800)	x	-
IMPNFIL		x	-

(Teil 1 von 2)

Makroaufruf	erforderliche VERSION-Angabe	MF-Operand	„DSECT-Makro“
IMPORT	ohne Operand VERSION	-	DMAIMP
	VERSION=1	x	-
LFFSNAP		x	-
LJFSNAP		x	-
MAILFIL		x	-
RDTFT	ohne Operand VERSION	-	DMARD (PLIST=INPUT) DMADR (PLIST=OUTPUT)
	VERSION≥2	x	-
REL	*(Makro ersetzt durch RELTFT)*	-	IDREL
RELTFT		x	-
REMPLNK		x	-
RFFSNAP		x	-
RJFSNAP		x	-
SHOPLNK		x	-
SHOPOOL		x	-
SHOWAIX		x	-
VERIF		-	IDVRF

(Teil 2 von 2)

x DSECT kann über MF-Operand erzeugt werden

\- kein MF-Operand/kein DSECT-Makro

Steuerblöcke und zugriffsmethodenspezifische Makroaufrufe

Makroaufruf	Kurzbeschreibung
IDECB	für den UPAM-Ereigniskennung-Steuerblock (FECB)
IDFCB	für den Dateisteuerblock (FCB) des Benutzerprogramms auf der TU-Ebene
IDFCBE	Erweiterung des 24-Bit-TU-FCB
IDMCB	EAM-Steuerblock für EAM-Makroaufruf
IDPPL	Operandenliste für den PAM-Makroaufruf
IDOST	Operandenliste für den OSTAT-Makroaufruf

DVS-Tabellen, Dateikatalog usw.

Makroaufruf	Kurzbeschreibung
IDCE	Katalogeintrag
IDCEG	Katalogeintrag (Zusatz für Dateigenerationsgruppen)
IDCEX	Katalogeintrag (Erweiterung)
IDEE	Katalogeintrag (Extent-Liste)
IDEMS	DVS-Fehlermeldungen
IDTFT	TFT-Eintrag
IDVT	Eintrag des Datenträger-Kennsatzes (Volume Table)

2.2.7 Ausgabe von Informationen über Dateien, Datenträger, Geräte usw.

Mit verschiedenen DVS-Makros lassen sich im Programm Informationen über Katalogeinträge und Zustand von Dateien, die Task File Table, die Belegung von Geräten und Datenträgern usw. anfordern und für die weitere Verarbeitung auswerten.

Makro	Kurzbeschreibung
FSTAT	Informationen aus dem Dateikatalog bzw. über Katalogeinträge von Dateien
OSTAT	Informationen über Zahl und Art der Zugriffe verschiedener Aufträge auf eine ISAM-Datei
RDTFT	Informationen über TFT-Einträge
SHOWAIX	Informationen über Sekundärschlüssel einer ISAM-Datei
SHOPLNK	Informationen über ISAM-Pool-Kettungsnamen
SHOPOOL	Informationen über ISAM-Pools

2.3 Gegenüberstellung von DVS-Makros und DVS-Kommandos

Makros und Kommandos mit gleicher Funktionalität

Makro	Kommando	Funktion
ADDPLNK	ADD-ISAM-POOL-LINK	Poolkettungsnamen für einen ISAM-Pool definieren
CATAL	CREATE-FILE	Katalogeintrag erstellen
	CREATE-FILE-GROUP	Dateigenerationsgruppen definieren
	MODIFY-FILE-ATTRIBUTES	Schutzmechanismen definieren
	MODIFY-FILE-GROUP-ATTRIBUTES	Schutzmechanismen definieren
CHNGE	CHANGE-FILE-LINK	Dateikettungsnamen in der TFT ändern
COMPFIL	COMPARE-DISK-FILES	Zwei Plattendateien vergleichen
COPFILE	COPY-FILE	Datei kopieren
CREAIX	CREATE-ALTERNATE-INDEX	Sekundärindex für ISAM-Datei erzeugen
CREPOOL	CREATE-ISAM-POOL	ISAM-Pool anlegen
DECFILE	DECRYPT-FILE	Verschlüsselte Datei entschlüsseln
DELAIX	DELETE-ALTERNATE-INDEX	Sekundärindizes einer ISAM-Datei löschen
DELPOOL	DELETE-ISAM-POOL	ISAM-Pool löschen/freigeben
DROPTFT	UNLOCK-FILE-LINK	Sperre für TFT-Eintrag aufheben
ENCFILE	ENCRYPT-FILE	Unverschlüsselte Datei verschlüsseln
ERASE	DELETE-FILE	Datei(en) löschen
	DELETE-FILE-GENERATION	Dateigenerationen löschen
	DELETE-FILE-GROUP	Dateigenerationsgruppe(n) löschen
	DELETE-SYSTEM-FILES	Systemdateien löschen
	EXPORT-FILE	Datei(en) exportieren
	EXPORT-NODE-FILE	Node-File(s) exportieren
FILE	ADD-FILE-LINK	TFT-Eintrag erstellen
	CREATE-FILE	Datei erzeugen
	CREATE-FILE-GENERATION	Dateigeneration erzeugen
	CREATE-TAPE-SET	TST-Eintrag erstellen
	EXTEND-TAPE-SET	TST-Eintrag erweitern
	IMPORT-FILE	Datei importieren

(Teil 1 von 2)

Makro	Kommando	Funktion
FILE (Forts.)	MODIFY-FILE-ATTRIBUTES	Merkmale einer Datei ändern
	MODIFY-FILE-GENERATION-SUPPORT	Merkmale einer Dateigeneration ändern
FSTAT	IMPORT-FILE	Informationen aus dem Dateikatalog zur Verfügung stellen
	SHOW-FILE-ATTRIBUTES	
IMPNFIL	IMPORT-NODE-FILE	Katalogeintrag für Node-Files erstellen (importieren)
IMPORT	CHECK-IMPORT-DISK-FILE	Importieren von Dateien vorab prüfen
	IMPORT-FILE	Katalogeintrag für Dateien erstellen (importieren)
LFFSNAP	LIST-FILE-FROM-SNAPSET	Dateien von einem Snapset auflisten
LJFSNAP	LIST-JV-FROM-SNAPSET	Jobvariablen von einem Snapset auflisten
MAILFIL	MAIL-FILE	Datei oder Bibliothekselement per E-Mail an eine Benutzerkennung versenden
RDTFT	SHOW-FILE-LINK	Informationen aus der TFT zur Verfügung stellen
RELTFT	DELETE-TAPE-SET	TST-Eintrag löschen
	REMOVE-FILE-LINK	TFT-Eintrag aufheben
REMPLNK	REMOVE-ISAM-POOL-LINK	Poolkettungsnamen löschen
RFFSNAP	RESTORE-FILE-FROM-SNAPSET	Dateien von einem Snapset restaurieren
RJFSNAP	RESTORE-JV-FROM-SNAPSET	Jobvariablen von einem Snapset restaurieren
SHOPLNK	SHOW-ISAM-POOL-LINK	Zuordnung von ISAM-Pools zu ISAM-Pool-Kettungsnamen ausgeben
SHOPOOL	SHOW-ISAM-POOL-ATTRIBUTES	Informationen über einen ISAM-Pool ausgeben
SHOWAIX	SHOW-INDEX-ATTRIBUTES	Informationen über Sekundärschlüssel einer ISAM-Datei ausgeben
VERIF	CHECK-FILE-CONSISTENCY	Dateikonsistenz wiederherstellen
	REMOVE-FILE-ALLOCATION-LOCKS	
	REPAIR-DISK-FILES	

(Teil 2 von 2)

DVS-Kommandos ohne entsprechenden Makroaufruf

Kommando	Funktion
ADD-CRYPTO-PASSWORD	hinterlegt das Crypto-Kennwort zur Entschlüsselung von verschlüsselten Dateiinhalten in der Kennworttabelle der Task
ADD-PASSWORD	Kennwort in die Kennworttabelle des Auftrags eintragen
CONCATENATE-DISK-FILES	SAM-Dateien verketten
EDIT-FILE-ATTRIBUTES EDIT-FILE-GROUP-ATTRIBUTES EDIT-FILE-GENERATION-SUPPORT	aktiviert den geführten Dialog des entsprechenden MODIFY-Kommandos und ermöglicht das „Editieren“ eines bestehenden Katalogeintrags
EDIT-FILE-LINK	aktiviert den geführten Dialog des ADD-FILE-LINK-Kommandos und ermöglicht das „Editieren“ eines bestehenden TFT-Eintrags
LIST-NODE-FILES	informiert über Node-Files auf dem Net-Storage
LOCK-FILE-LINK	TFT-Eintrag sperren, sodass er erst nach einem UNLOCK-FILE-LINK-Kommando freigegeben wird.
REMOVE-CRYPTO-PASSWORD	entfernt das Crypto-Kennwort aus der Kennworttabelle der laufenden Task
REMOVE-PASSWORD	Kennwort aus der Kennworttabelle des Auftrags löschen
REPAIR-FILE-LOCKS	unberechtigt sitzende Dateisperren beseitigen
RESTART-PROGRAM	Programm an einem Punkt starten, der mit WRCPT- (oder CHKPT-) Makroaufruf gesichert wurde.
SHOW-BLOCK-TO-FILE-ASSIGNMENT	privilegiertes Kommando: Anzeigen von Dateien, in denen die angeforderten Blöcke liegen
SHOW-FILE	Ausgabe einer Datei auf SYSOUT
SHOW-FILE-LOCKS	Sperren einer Datei anzeigen
START-FILE-CACHING	Caching von bereits geöffneten Dateien starten
STOP-FILE-CACHING	Caching einzelner Dateien beenden

3 Hinweise zur Programmierung

In diesem Kapitel sind programmierrelevante Besonderheiten der verschiedenen Zugriffsmethoden dargestellt.

3.1 BTAM - Basic Tape Access Method

BTAM ist eine Zugriffsmethode für blockorientierte Bandverarbeitung; mit BTAM lassen sich auch Banddateien verarbeiten, die nicht mit BTAM erstellt wurden. Während der Verarbeitung einer Banddatei kann die Verarbeitungsrichtung innerhalb der Datei beliebig gewechselt werden. Die Bänder können beliebig block- oder abschnittsweise positioniert werden. BTAM verarbeitet Dateien mit oder ohne Standardblockung.

Makroaufrufe für die Zugriffsmethode BTAM

Folgende Makroaufrufe können von der Zugriffsmethode BTAM genutzt werden:

Makro	Operand	Funktion
CLOSE		Service-Makros
EXLST		
EXRTN		
FCB		
FCBAD		
IDFCB		
IDFCBE		
LBRET		
OPEN		

(Teil 1 von 2)

Makro	Operand	Funktion
BTAM	CHK	Verarbeitungszustand einer Ein-/Ausgabeoperation feststellen
	ERG	Blocklücke erzeugen
	MINF	Medium-Information holen (nur sinnvoll für Volume-Typen, die optische Platten beinhalten)
	POS	Band positionieren
	RBID	Bandposition bestimmen
	RD/ RDWT	Daten in den Hauptspeicher lesen und den Abschluss der Ein-/Ausgabe-operation abwarten
	RDBF	Daten aus dem Sicherstellungsbereich des MBK-Puffers lesen
	REV/ REVWT	Band rückwärts lesen und den Abschluss der Ein-/Ausgabeoperation abwarten
	RT/ RTL	Lesen mit Datentransfer; mit/ohne Meldung bei kürzerer Länge als erwar-tet
	RNT/ RNTL	Lesen ohne Datentransfer; mit/ohne Meldung bei kürzerer Länge als erwartet
	SYNC	Synchronisieren und Bandposition bestimmen
	WRT/WR TWT	Daten aus dem Hauptspeicher schreiben / und den Abschluss der Ein-/Ausgabeoperation abwarten
	WT	Warten auf Abschluss der Ein-/Ausgabeoperation
	BSF	Synchronisieren und Bandposition bestimmen
	BSR	Steuerschlüssel für das Positionieren und für das Schreiben von Abschnittsmarken
	FSF	
	FSR	
	REW	
	RUN	
	WTM	

(Teil 2 von 2)

OPEN-Modi

Folgende Open-Modi sind bei der Zugriffsmethode BTAM möglich:

INOUT	Auffinden von Sätzen in einer vorhandenen Datei sowie Hinzufügen von Sätzen; es werden keine Anfangskennsätze erstellt, da die Datei bereits vorhanden sein muss.
INPUT	Auffinden von Sätzen in einer vorhandenen Datei in Richtung Dateiende.
OUTIN	Erstellen einer neuen Datei und/oder Auffinden von Sätzen in der Datei; es werden Kennsätze erstellt, da eine neue Datei eingerichtet wird.
OUTPUT	Erstellen einer neuen Datei.
REVERSE	Wie INPUT, das Band wird jedoch beim OPEN auf das Dateiende positioniert. Dateien, die sich über mehrere Datenträger erstrecken, lassen sich nur einzeln (mit VSEQ) verarbeiten.
SINOUT	Wie INOUT, das Band wird jedoch nicht positioniert; die Angabe ist unzulässig, wenn auf Bandanfang positioniert ist.

Die OPEN-Modi INPUT und REVERSE unterscheiden sich zum OPEN-Zeitpunkt nur in der Bandpositionierung. Ein OPEN REVERSE mit anschließender RD- oder RDWT-Operation führt nicht zum Lesen vom Bandende in Richtung Bandanfang.

BTAM-Operationen und OPEN-Modi

OPEN-Typ **BTAM-Operation**	**INPUT**	**REVERSE**	**OUTPUT**	**INOUT / OUTIN / SINOUT**
CHK	x	x	x	x
Control*	n	n	x	x
ERG			x	x
RD/RDWT	x	x		x
REV/REVWT	x	x		x
RT/RTL	x	x		x
RNT/RNTL	x	x		x
WRT/WRTWT			x	x
WT	x	x	x	x
MINF	x	x	x	x
POS	x	x	x	x

(Teil 1 von 2)

OPEN-Typ BTAM-Operation	INPUT	REVERSE	OUTPUT	INOUT / OUTIN / SINOUT
RBID	x	x	x	x
RDBF			x	x
SYNC	x	x	x	x

(Teil 2 von 2)

x = zulässig

n = Ausgabe-Steuerfunktionen sind nicht zulässig

* „Control“ steht für die Operanden FSF, BSF, WTM, RUN, ERG, FSR, BSR, REW. Sie werden beim Makro „BTAM“ (Seite 117) beschrieben.

BTAM-Satz-/Blockformate

BTAM ist eine blockorientierte Zugriffsmethode für Banddateien des Formats BLKCTRL= NO. Für die Blocklänge gilt: 18 Byte ≤ Blocklänge ≤ 32768 Byte (vgl. Beschreibung des Operanden LEN im BTAM-Aktionsmakroaufruf).

BTAM wertet den FCB-Operanden RECFORM aus. Die Angabe des Satzformates wird auf die Blocklänge bezogen.

– RECFORM=F für Blöcke fester Länge
  Die Länge wird mit BLKSIZE=länge definiert oder ist als LEN-Angabe beim BTAM-Makroaufruf angegeben. Für die minimale/maximale Länge gelten die oben angegebenen Werte.

– RECFORM=U für Blöcke undefinierter Länge
  BTAM entnimmt die Blocklänge entweder dem LEN-Operanden beim Makroaufruf oder dem Register, das in FILE/FCB mit RECSIZE=reg spezifiziert wurde.

– RECFORM=V für Blöcke variabler Länge
  (Satzformat V wird behandelt wie Satzformat U)

Mit SAM erstellte Banddateien können mit BTAM blockweise gelesen werden. Da BTAM die Satzstruktur nicht kennt, muss der Anwender für das Entblocken der Sätze selbst sorgen.

3.2 DIV - Data In Virtual

Data in Virtual (DIV) ist eine Zugriffsmethode, die sich von den traditionellen Zugriffsmethoden wie ISAM, SAM oder UPAM dadurch unterscheidet, dass sie ohne eine Strukturierung der Datei in Sätze oder Blöcke, ohne I/O-Puffer und ohne Operationen wie GET oder PUT auskommt.

DIV arbeitet mit einem speziellen DIV-OPEN.

DIV ist eine objektorientierte Zugriffsmethode, insbesondere geeignet zur Bearbeitung unstrukturierter Daten (Binary Large Objects (BLOBS)).

Bei der DIV-Schittstelle handelt es sich um eine SVC-Schnittstelle. Die Aufträge werden durch eine Parameterliste formuliert; Rückmeldungen über das Ergebnis erfolgen über einen Returncode in der Parameterliste (nicht über Exits).

Makroaufrufe für die Zugriffsmethode DIV

Die Zugriffsmethode DIV arbeitet mit dem Makroaufruf DIV, der die verschiedenen Funktionen zur Dateiverarbeitung umfasst:

Makro	Operand	Funktion
DIV	CLOSE	Datei schließen; ggf. noch existierende Fenster werden mit Standardwerten freigegeben
	MAP	Fenster (Arbeitsbereich im virtuellen Adressraum) definieren
	OPEN	Datei eröffnen
	RESET	Änderungen im Fenster zurücknehmen
	SAVE	Modifizierte Seiten aus dem Fenster in die Datei auf Platte zurückschreiben
	UNMAP	Fenster im virtuellen Adressraum freigeben

Datei eröffnen

Mit der Funktion OPEN wird einem Programm die Bearbeitung einer Datei erst ermöglicht. Die Funktion OPEN prüft u.a., ob ein Benutzer die erforderliche Zugriffsberechtigung besitzt, ob eine Datei bereits eröffnet ist und ob die Eröffnungsmodi miteinander verträglich sind. Die Öffnungsmodi und der Zugriff auf die Datei durch mehrere Benutzer entsprechen im Wesentlichen der OPEN-Funktion für die anderen DVS-Zugriffsmethoden (siehe Handbuch „Einführung in das DVS“ [1]).
Für die Datei kann entweder nur Lesezugriff oder sowohl Lese- als auch Schreibzugriff vereinbart werden. Bei Schreibzugriff kann die Datei modifiziert, erweitert oder von Dateianfang an neu beschrieben werden. Sowohl ein Schreiber und/oder mehrere Leser als auch mehrere Schreiber/Leser können auf die Datei gleichzeitig zugreifen.
Der Anwender kann festlegen, ob ein Dateibereich bzw. die gesamte Datei sofort bei Definition eines Fensters in das Fenster eingelesen werden soll oder ob eine angegebene Seite erst beim ersten Zugriff auf sie in das Fenster eingelesen werden soll.

Wichtige Operanden bei Datei-Eröffnung

Operand	Operanden-wert	Bedeutung
FCT	*OPEN	Datei eröffnen
LARGE_ FILE		Mit dem Funktionsoperanden LARGE_FILE wird angegeben, ob die zu eröffnende Datei eine „große Datei“ werden soll, deren Dateigröße ≥ 32 GB werden kann.
	*ALLOWED	Die Datei darf eine große Datei werden.
	*FORBIDDEN	Die Datei darf keine große Datei werden.
MODE		Mit dem Funktionsoperanden MODE wird angegeben, welche Verarbeitung (Lesen, Schreiben) für die Datei vorgesehen ist (Schreiben bedeutet die Ausführung der DIV-Funktion SAVE).
	*INPUT	Die Datei wird als Eingabedatei eröffnet; die Ausführung der Funktion SAVE ist nicht erlaubt.
	*INOUT	Die Datei muss existieren; es sind sowohl Lese- als auch Schreiboperationen zulässig.
	*OUTIN	Die Datei wird neu erstellt; es sind sowohl Lese- als auch Schreiboperationen zulässig.
SHARUPD		Mit dem Funktionsoperand SHARUPD wird in Abhängigkeit von MODE angegeben, welche Multi-User Betriebsarten erlaubt sein sollen (vgl. hierzu Tabellen im Abschnitt Multi-User-Betrieb):
	*NO	Ein Schreiber **oder** mehrere Leser können die Datei gleichzeitig eröffnet haben.
	*WEAK	Ein Schreiber **und** mehrere Leser können die Datei gleichzeitig eröffnet haben.
	*YES	**mehrere** Schreiber können die Datei gleichzeitig eröffnet haben.
LOCVIEW		Mit dem Operanden LOCVIEW wird festgelegt, zu welchem Zeitpunkt eine Seite in ein Fenster eingelesen wird.
	*MAP	Bereits bei der Definition eines Fensters (FCT=*MAP) werden alle Seiten des angegebenen Dateibereiches eingelesen.
	*NONE	Erst beim Zugriff auf eine Seite wird die Seite aus der Plattendatei in das Fenster eingelesen.

Multi-User-Betrieb

Eine UPAM-Datei kann mit den Zugriffsmethoden UPAM (siehe Seite 89), FASTPAM (siehe Seite 61) oder DIV erstellt bzw. bearbeitet werden. FASTPAM und DIV können jedoch nur UPAM-Dateien mit der Eigenschaft BLKCTRL=NO bearbeiten.)

Die Erlaubnis für eine parallele Dateibearbeitung ist von den bei der Eröffnung angegebenen Operandenwerten für SHARUPD, MODE, LOCKENV und LOCVIEW abhängig.

Die möglichen parallelen Eröffnungen werden in der folgenden Tabelle dargestellt:

Verträglichkeits-Matrix bei DIV-OPEN

			USER B									
			SHARUPD =									
			*YES			*NO			*WEAK			
	OPEN-Modus		INPUT	INOUT	OUTIN	INPUT	INOUT	OUTIN	INPUT	LMAP	INOUT	OUTIN
USER A	SHARUPD =*YES	INPUT	X	O		X			X	X		
		INOUT	O	O					X			
		OUTIN	O	O					X			
	SHARUPD =*NO	INPUT	X			X			X	X		
		INOUT							X			
		OUTIN							X			
	SHARUPD =*WEAK	INPUT	X	X		X	X		X	X	X	
		LMAP	X			X			X	X	O	
		INOUT							X	O		
		OUTIN							X			

LMAP: INPUT LOCVIEW=MAP (gibt es nur bei DIV)

X: OPEN erlaubt

O: OPEN nur erlaubt, wenn die Eröffner dieselbe blockorientierte Zugriffsmethode benutzen (nur UPAM/FASTPAM oder nur DIV)

und denselben Wert für den Operanden LOCKENV benutzen (alle LOCKENV=*HOST oder LOCKENV=*XCS)

und alle im selben HOST laufen *oder* in einem XCS-Verbund bei Verwendung von LOCKENV=*XCS

Anmerkungen

– Leseoperationen mit SHARUPD=*WEAK können eine Datei gleichzeitig mit jeder beliebigen Schreiboperation eröffnet haben.

  Den Leseoperationen mit DIV-SHARUPD=*WEAK, die bei OPEN LOCVIEW=*MAP spezifiziert haben, werden bereits bei MAP alle Fensterseiten aus der Datei in das Fenster eingelesen, wobei von DIV sichergestellt wird, dass während des Einlesens keine Dateiseiten durch einen parallelen SAVE einer Schreiboperation mit DIV-SHARUPD=*WEAK verändert werden können.
  Bei einer UPAM/FASTPAM-Schreiboperation existiert dieser Schutz gegen paralleles Schreiben nicht.

  Aus diesem Grund sind Leseoperationen mit DIV-SHARUPD=*WEAK, für die LOCVIEW=*MAP spezifiziert wurde, nur mit Schreiboperationen mit DIV-SHARUPD=*WEAK verträglich, nicht jedoch mit anderen Schreiboperationen.

  Auch die übrigen oben für den Eintrag 'O' formulierten Bedingungen müssen erfüllt sein (alle Eröffner mit gleichen Werten für den LOCKENV-Operanden und alle Eröffner im gleichen Host oder - wenn in verschiedenen Hosts - mit der Angabe LOCKENV=*XCS).

  Leseoperationen mit DIV-SHARUPD=*WEAK, die bei OPEN LOCVIEW=*NONE spezifiziert haben, besitzen dieselbe Verträglichkeit wie Leseoperationen mit UPAM/FASTPAM-SHARUPD=*WEAK.

– Eröffner mit DIV-SHARUPD=*YES sind nicht mit Eröffnern mit UPAM/-FASTPAM-SHARUPD=*YES verträglich.

– Leseoperationen sind immer miteinander verträglich (unabhängig von Zugriffsmethode, SHARUPD-Spezifikation, LOCKENV-Spezifikation und Host).

– SHARUPD=*YES:
  Bei jedem Aufruf des Allocators wird die Dateigröße überprüft.
  Wenn bei dieser Überprüfung eine Dateigröße ≥ 32 GB ermittelt wird und im zugehörigen FCB das Attribut LARGE_FILE=*FORBIDDEN bzw. in der TFT das Attribut EXCEED-32GB=*FORBIDDEN gesetzt ist, wird die Verarbeitung abgebrochen.
  DIV liefert in diesem Fall den Returncode `X'00400030'` in seiner eigenen Parameterliste DIV(I).

Fenster definieren

Mit der Funktion MAP wird ein Fenster definiert. Das Fenster ist ein Bereich in einem Adressraum (Programmraum oder Datenraum), der einem Dateibereich oder einer ganzen Datei zugeordnet ist.
Bei OPEN wird festgelegt, ob eine Fensterseite erst beim ersten Zugriff oder bereits zum MAP-Zeitpunkt in das Fenster eingelesen wird. Über die einzelnen Operanden der Funktion MAP werden Art (Programmraum oder Datenraum), Lage und Größe des virtuellen Adressraums festgelegt.

Mit den Funktionen FCT=*MAP und FCT=*UNMAP des Makros DIV können Fenster für einen Programmraum auf- und abgebaut werden. Für den Operanden SPID (Kennung des Datenraumes) muss stets SPID=0 angegeben werden.

Folgende Regeln sind zu beachten:

– Eine Seite eines Adressraums darf nur **einem** Fenster zugeordnet sein
– Eine Dateiseite darf innerhalb der Fenster eines OPEN nur **einem** Fenster zugeordnet sein.
– Eine Dateiseite darf mehreren Fenstern zugeordnet sein, wenn diese verschiedenen OPEN angehören.

Wichtige Operanden bei Fenster-Definition

Operand	Operanden-wert	Bedeutung
FCT	*MAP	Fenster definieren
SPID		Mit SPID wird die Kennung des Datenraums angegeben, in dem das Fenster angelegt werden soll.
AREA		Mit AREA wird die Anfangsadresse des Fensters in einem virtuellen Adressraum (auf 4-KByte-Seitengrenze) angegeben. Eine Seite eines Adressraums darf nur **einem** Fenster zugeordnet sein.
OFFSET SPAN		Mit den Operanden OFFSET und SPAN wird der Dateibereich in Einheiten von 4-KB-Seiten (Anfang und Länge) und damit die Größe des Fensters festgelegt. Über die Funktion MAP kann eine Datei physikalisch verlängert werden.
DISPOS	*OBJECT *UNCHNG	Mit DISPOS kann festgelegt werden, welchen Inhalt das Fenster haben soll: Beim ersten Zugriff auf eine Seite steht die entsprechende Seite der Datei im Fenster. Eine Fensterseite wird beim ersten Zugriff nicht durch die ihr entsprechende Dateiseite ersetzt, sondern behält ihren Inhalt (Inhalt des virtuellen Adressraums). *Hinweis* Nach einem REQM- oder DSPSRV-Makro ist der angegebene virtuelle Adressraum mit X' 00' initialisiert (siehe Handbuch „Einführung in das DVS“ [1])

Zurückschreiben in die Plattendatei

Mit der Funktion SAVE werden modifizierte Fensterseiten in die Plattendatei zurückgeschrieben (gesichert).

Es wird ein Dateibereich angegeben, für den die Funktion SAVE ausgeführt werden soll. Sämtliche geänderten Fensterseiten dieses Bereichs werden in die Plattendatei zurückgeschrieben. Dabei kann die Datei **logisch** verlängert oder verkürzt werden. (Physikalisch kann die Datei nur durch die Funktion MAP verlängert werden, siehe Handbuch „Einführung in das DVS“ [1]). Die neue Dateilänge wird durch die Funktion SAVE in einem Feld der Parameterliste (DIVPSIZE) zur Verfügung gestellt.

Wichtige Operanden beim Zurückschreiben in die Plattendatei

Operand	Operanden-wert	Bedeutung
FCT	*SAVE	Zurückschreiben in die Plattendatei
OFFSET SPAN		Mit den Operanden OFFSET und SPAN wird der Dateibereich festgelegt, für den die Funktion SAVE ausgeführt werden soll (Anfang; Länge in 4-KB-Blöcken). Dabei kann die Datei **logisch** verlängert oder verkürzt werden. die neue Dateilänge wird durch SAVE zur Verfügung gestellt. (Physikalisch kann die Datei nur durch die Funktion MAP verlängert werden; siehe Dateiverlängerung im Handbuch „Einführung in das DVS“ [1]).

Zurücknehmen von Modifikationen im Fenster

Mit der Funktion RESET können Änderungen in Fenstern zurückgenommen („gelöscht“) werden, die seit MAP oder dem letzten vorausgegangenen SAVE durchgeführt wurden. Es wird ein Dateibereich festgelegt, für den die Funktion RESET ausgeführt werden soll. Hierbei ist die Abhängigkeit vom Operanden DISPOS der Funktion MAP zu beachten:

– Ist ein Fenster mit DISPOS=*OBJECT definiert, so erscheint bei einem nach RESET folgenden Zugriff auf eine Seite, die entsprechende Seite der Datei im Fenster.
– Ist ein Fenster mit DISPOS=*UNCHNG definiert, so ist bei einem nach RESET folgenden Zugriff auf eine Seite diese Seite mit X'00' initialisiert, es sei denn die Seite wurde schon einmal seit Eröffnung des Fensters durch SAVE in die Datei geschrieben. In diesem Fall wird sie nach RESET ebenfalls aus der Datei gelesen.

Wichtige Operanden beim Zurücknehmen von Modifikationen im Fenster

Operand	Operanden-wert	Bedeutung
FCT	*RESET	Zurücknehmen von Modifikationen im Fenster
OFFSET SPAN		Mit den Operanden OFFSET und SPAN wird der Dateibereich festgelegt, für den die Funktion RESET ausgeführt werden soll (Anfang; Länge in 4-KB-Blöcken). Alle Modifikationen in den Fensterseiten des RESET-Bereichs werden zurückgenommen.
RELEASE		Mit dem Operanden RELEASE=*YES erstreckt sich die RESET-Funktion auf **alle** Seiten des angegebenen Bereichs - nicht nur auf modifizierte Seiten.

Fenster abbauen

Mit UNMAP wird ein Fenster abgebaut. Der Anwender kann angeben, mit welchem Inhalt die Seiten des (ehemaligen) Fensters nach Ausführen der Funktion UNMAP verbleiben sollen. Entweder die Fensterseiten werden mit X'00' initialisiert oder sie erscheinen dem Benutzer so wie vor UNMAP.

Wichtige Operanden beim Abbauen von Fenstern

Operand	Operanden-wert	Bedeutung
FCT	*UNMAP	Fenster abbauen
SPID		Gibt den Adressraum an, in dem das Fenster liegt.
AREA		Gibt die Anfangsadresse des Fensters an, das innerhalb des mit SPID angegebenen Adressraums liegt.
DISPOS	*FRESH *UNCHNG	Angabe in welchem Zustand die Seiten des (ehemaligen) Fensters verbleiben sollen. Die Fensterseiten werden initialisiert (X' 00'). Die Fenster behalten aus Sicht des Programms den Inhalt, den sie bereits vor der Funktion UNMAP hatten.

Datei schließen

Mit der Funktion CLOSE wird eine angegebene Datei geschlossen. Eventuell noch vorhandene Fenster werden mit den Standardwerten abgebaut.

Wichtige Operanden bei Datei-Schließung

Operand	Operanden-wert	Bedeutung
FCT	*CLOSE	Datei schließen.
ID		Kennung der Datei, die geschlossen werden soll.

3.3 EAM - Evanescent Access Method

Mit EAM werden taskspezifische Dateien im SYSEAM-Bereich verarbeitet. EAM ist eine blockorientierte Zugriffsmethode und eignet sich vor allem für die schnelle Verarbeitung von auftragsabhängigen Arbeitsdateien.

Makroaufruf für die Zugriffsmethode EAM

Sämtliche EAM-Funktionen werden durch den Makro EAM gesteuert. Der Makro EAM umfasst folgende Funktionen:

Makro	Funktion
EAM	– Einrichten und Eröffnen einer neuen Datei – Eröffnen einer bestehenden Datei – Lesen (blockweise, sequenziell oder direkt) – Schreiben (blockweise, sequenziell oder direkt) – Prüfen und Warten auf Beendigung der Ein-/Ausgabe – Schließen einer Datei – Löschen einer Datei

Eine Operation wird durch die Angabe eines sedezimalen Operationsschlüssels im MFCB ausgewählt und mit dem EAM-Makroaufruf angestoßen. Die Wirkung ist durch die MFCB-Felder bestimmt, die EAM nach Analyse des Operationsschlüssels zusätzlich auswertet (siehe Tabelle auf Seite 49).

Der Makro EAM steuert alle EAM-Zugriffe. EAM hat folgende Eigenschaften:

– EAM-Dateien werden nicht katalogisiert. Das Eröffnen einer EAM-Datei erfordert deshalb keinen Plattenzugriff.

– Jede EAM-Datei wird nach Beendigung des sie eröffnenden Auftrags automatisch gelöscht (temporäre Datei).

– Die Verständigung zwischen EAM und dem Anwender erfolgt nur über den EAM-Steuerblock (MFCB = Mini File Control Block). Modifizierung des MFCB zum Eröffnungszeitpunkt ist nicht vorgesehen.

– EAM benutzt nur gemeinschaftliche Datenträger (Pubsets). Dabei spielt es keine Rolle, ob es sich um Platten mit oder ohne PAM-Schlüssel (K- oder NK-Platten) handelt.

– Der Speicherplatzbedarf für die EAM-Routinen und die Laufzeiten für die Lese- und Schreibzugriffe sind geringer als bei den Zugriffsmethoden für katalogisierte Dateien.

– Eine EAM-Datei kann nur von dem Auftrag bearbeitet werden, der sie angelegt und eröffnet hat. Ein Auftrag kann mehrere EAM-Dateien gleichzeitig eröffnen und bearbeiten.

– EAM ist eine blockorientierte Zugriffsmethode, Grundlage der Verarbeitung ist ein 2048 Byte langer Block (= PAM-Seite). Bei geketteter Ein-/Ausgabe können bis zu 16 aufeinander folgende Blöcke mit einem Makroaufruf bearbeitet werden.

– EAM-Dateien können die Größe von 32 GB nicht überschreiten.

– Bei Programm-Wiederanlauf mit RESTART-PROGRAM werden alle EAM-Dateien des Auftrags gelöscht.

Hinweis

Wo die EAM-Datei angelegt wird, ist abhängig davon, ob Shared-Pubsets eingesetzt werden und welche Festlegungen die Systembetreuung getroffen hat. Die genaue Beschreibung befindet sich im Handbuch „Einführung in die Systembetreuung“ [7].

MFCB (Mini File Control Block)

Aufbau des MFCB

Die Feldeinteilung des MFCB und die Auswertung der MFCB-Felder in Abhängigkeit von der gewählten Operation sind aus der folgenden Tabelle zu ersehen.

Operations-schlüssel	MFCB-Felder										
	Funk-tions-einheit	**Ver-sionsan-gabe**	**Return-Code**	**Zusatz-byte**	**Logische Block-nummer**	**Datei-name**	**Fehler-Byte**	**Status-Byte**	**Adresse IOAREA1 IDMFIO1**	**Anzahl zu übertr. Blöcke**	**Adresse IOAREA2 IDMFIO2**
IDMFOPC	IDM FUNIT*	IDM VERS*	IDM RETCO*	IDMFOC	IDMFLBN	IDMFFN	IDMFEB	IDMFSB	IDMFIO A1*	IDMFNHP	IDMFIO A2*
Eröffnen neue Datei (IDMFO)	A	A	S	A	(+)	S	S	-	-	(A)	-
Wiedereröff-nen (IDMFRO)	A	A	S	A	S	A(+)	S	-	-	(A)	-
Lesen (IDMFRD)	A	A	S	A	A	A	S	S	(A)	-	(A)
Schreiben (IDMFWR)	A	A	S	A	A	A	S	S	(A)	-	(A)
Prüfen (IDMFCK)	A	A	S	-	-	A	S	S	-	-	-
Prüfen und Warten (IDMFCW)	A	A	S	-	-	A	S	S	-	-	-
Datei schlie-ßen (IDMFCL)	A	A	S	-	S	A	S	S	-	-	-
Datei lö-schen (IDMFER)	A	A	S	-	-	A	S	-	-	-	-

* gilt nur für PARMOD=31

A Feldinhalt wird ausgewertet

S Feldinhalt wird vom System gesetzt

\+ Ausnahmen für Bindemoduldateien (siehe „Behandlung von Bindemoduldateien durch EAM“ auf Seite 60)

Beschreibung des MFCB

Die Feldeinteilung des MFCB und die Auswertung der MFCB-Felder in Abhängigkeit von der gewählten Operation sind aus der Tabelle auf Seite 49 zu ersehen.

Der MFCB ist der Verständigungsbereich zwischen EAM und Anwender; der MFCB muss auf Wortgrenze ausgerichtet sein. Vor Aufruf des EAM-Makros müssen die für die gewählte Operation benötigten Felder versorgt werden.

Der MFCB kann mit dem Makro IDMCB mit symbolischen Namen versehen werden.

Funktionseinheit (IDMFUNIT)

Gilt explizit oder implizit PARMOD=31, muss IDMFUNIT mit dem Wert „DMFEAM“ versorgt sein.

Operationsschlüssel (IDMFOPC)

IDMFO	Einrichten und Eröffnen einer neuen Datei (OPEN) - EAM wertet das Bit im Zusatzbyte IDMFOO aus und eröffnet danach entweder die Bindemoduldatei oder eine neue auftragsspezifische Datei. Der Name einer neuen Bindemoduldatei wird im TCB (Task Control Block) festgehalten; ist bereits eine Bindemoduldatei vorhanden, wird diese wieder eröffnet (d.h. für die Bindemoduldatei gilt OPEN=REOPEN; siehe „Behandlung von Bindemoduldateien durch EAM“ auf Seite 60). Einer neuen EAM-Datei wird ein binärer Dateiname zwischen 1 und 14000 zugewiesen und in das Feld „Dateiname“ geschrieben. Außerdem wird das Bit IDMFCI des Zusatzbytes ausgewertet. Falls dieses Bit gesetzt ist, d.h. im geketteten Ein-/Ausgabemodus, wird überprüft, ob das Feld IDMFNHP (= Anzahl der zu übertragenden Blöcke) eine Zahl zwischen 1 und 16 enthält. Die Adressen der Ein-/Ausgabebereiche werden nicht überprüft.

IDMFRO	Wiedereröffnen einer bestehenden Datei (REOPEN) - im Zusatzbyte werden die Bits IDMFOO, IDMFSBR und IDMFCI ausgewertet. Für IDMFCI und damit für das Feld IDMFNHP gilt das Gleiche wie bei der OPEN-Operation. Entsprechend der Auswertung von IDMFOO wird entweder die auftragsspezifische Bindemoduldatei oder eine im Feld „Dateiname“ (IDMFFN) genannte Datei eröffnet (siehe auch „Behandlung von Bindemoduldateien durch EAM“ auf Seite 60).
IDMFRD	Lesen (READ) - EAM wertet im Zusatzbyte das Bit IDMFI1 aus und überprüft die über dieses Bit ausgewählte Adresse eines Eingabebereichs (Feld IDMFIO1 oder IDMFIO2). Die Eingabe erfolgt an die entsprechende Adresse, auch wenn der Inhalt von IDMFIO1/2 unmittelbar nach Aufruf der Operation verändert wurde. Wird bei einem Lesevorgang das Dateiende erkannt, wird das Bit IDMFEF des Fehlerbytes gesetzt. War zum OPEN/REOPEN-Zeitpunkt das Bit IDMFCI des Zusatzbytes gesetzt, erfolgt die Eingabe gekettet. Die Anzahl der zu übertragenden Blöcke wird dem Feld IDMFNHP ebenfalls zum OPEN/REOPEN-Zeitpunkt entnommen (siehe „Anzahl zu übertragender Blöcke (IDMFNHP)“ auf Seite 55). Das Feld „Logische Blocknummer“ (IDMFLBN) enthält die Nummer des nächsten zu lesenden Blocks oder den Wert null für sequenzielles Lesen. Die Lese-/Schreiboperation ist bei EAM asynchron, d.h. der Benutzer erhält sofort nach Aufruf des EAM-Makros die Steuerung zurück, es sei denn, eine vorhergehende Ein-/Ausgabe-Operation ist noch nicht abgeschlossen. Im MFCB eingetragene diesbezügliche Fehlerinformationen (IDMFERR, IDMFSB) beziehen sich immer auf die vorhergehende Operation, daher muss bei unmittelbar aufeinander folgenden Ein-/Ausgabe-Operationen die Zweite warten. Überlappte Ein-/Ausgabe bzw. Doppelpufferung kann über die Angabe von zwei verschiedenen Ein-/Ausgabebereichsadressen erreicht werden (siehe „Überlappte Ein-/Ausgabe“ auf Seite 58).
IDMFWR	Schreiben (WRITE) - wie beim Lesen wird in Abhängigkeit vom Bit IDMFI1 entweder das Feld IDMFIO1 oder das Feld IDMFIO2 auf Gültigkeit der dort enthaltenen Adresse eines Ausgabebereichs überprüft. Ist der Inhalt des Feldes „Logische Blocknummer“ (IFMFLBN) null, wird sequenziell im Anschluss an das Dateiende geschrieben. Ist der Inhalt $\neq 0$ wird der zu übertragende Block an die so bezeichnete Stelle der Datei gebracht. Für den Ablauf einer Schreiboperation gilt im Übrigen das Gleiche wie für Leseoperationen (siehe oben).
IDMFCK	Prüfen der Beendigung einer Ein-/Ausgabeoperation (CHECK) - es wird überprüft, ob eine ausstehende Ein-/Ausgabeoperation abgeschlossen ist; der Benutzer erhält in jedem Fall die Steuerung sofort zurück. Ist die überprüfte Operation noch nicht abgeschlossen, erhält Register 15 den Wert 8. Ist die Operation beendet, werden die Statusbytes in den MFCB (IDMFSB) übertragen und Register 15 erhält den Wert 0.

IDMFCW	Prüfen einer Ein-/Ausgabe und Warten (CHECK WAIT) - die Beendigung der letzten Ein-/Ausgabeoperation wird abgewartet, anschließend werden die Statusbytes übertragen. Wurde die Übertragung der Statusbytes bereits durch eine andere Operation veranlasst, ist dieser EAM-Aufruf wirkungslos.
IDMFCL	Datei schließen (CLOSE) - nach Beendigung der letzten noch ausstehenden Ein-/Ausgabeoperation wird die Datei als geschlossen markiert. Die Blocknummer des letzten Blocks der Datei wird in das Feld „Logische Blocknummer“ (IDMFNLB) übertragen.
IDMFER	Datei löschen (ERASE) - die Datei wird gelöscht, gleichgültig, ob sie eröffnet ist oder nicht.

Versionsnummer (IDMVERS)

Gilt explizit oder implizit PARMOD=31, muss IDMVERS mit dem Wert DMEAMV versorgt sein. Dies ist wichtig im Hinblick auf zukünftige BS2000-Versionen, da so verschiedene Versionen dieser Schnittstelle ohne Neuübersetzung unterstützt werden können.

Return-Information (IDMRETCO)

In der 31-Bit-Version wird der Return-Code im Feld IDMRETCO des MFCB hinterlegt. Dieser Return-Code entspricht dem des Registers 15.

Return-Code	**Bedeutung**
0	Operation erfolgreich abgeschlossen
4	Operation nicht erfolgreich abgeschlossen Fehlerbytes (IDMFEB) überprüfen
8	nach Prüfoperationen: geprüfte Ein-/Ausgabe-Operation noch nicht abgeschlossen

Zusatzbyte (IDMFOC)

IDMFOO	Bindemoduldatei eröffnen - dieses Bit wird bei allen Operationen ausgewertet. Ist das Bit gesetzt, wird die auftragsspezifische Bindemoduldatei bearbeitet (siehe „Behandlung von Bindemoduldateien durch EAM“ auf Seite 60). Ist das Bit nicht gesetzt, wird in der OPEN-Funktion eine neue Datei eröffnet und der Dateiname in das Feld „Dateiname“ (IDMFFN) gebracht. In der Funktion REOPEN = Wiedereröffnen wird die in „Dateiname“ genannte Datei eröffnet.
IDMFCI	gekettete Ein-/Ausgabe - das Bit wird zum OPEN-/REOPEN-Zeitpunkt ausgewertet und sein Inhalt sichergestellt. Das heißt, ist beim (Wieder-)Eröffnen einer Datei dieses Bit gesetzt, erfolgen spätere Ein-/Ausgabeoperationen gekettet. Die Länge der Übertragungskette ist durch den Inhalt des Feldes IDMFNHP (=Anzahl der zu übertragenden Blöcke) festgelegt, das ebenfalls beim Öffnen der Datei ausgewertet wird. Ist das Bit nicht gesetzt, wird keine gekettete Ein-/Ausgabe verwendet.
IDMFSBR	Startpunkt der Ein-/Ausgabe in der Datei - über dieses Bit wird gesteuert, wo die Ein-/Ausgaben in der Datei „aufsetzen“ sollen. Das Bit wird beim Wiedereröffnen einer Datei ausgewertet. Ist es gesetzt, wird der Wert 0 in das Feld „Logische Blocknummer“ gebracht, sonst die höchste bisher vergebene Blocknummer der Datei (siehe Beschreibung des Feldes IDMFLBN, Seite 53).
IDMFI1	Steuerung des Ein-/Ausgabebereichs - dieses Bit wird bei aktuellen Ein-/Ausgabeoperationen ausgewertet. Ist das Bit gesetzt, wird die im Feld IDMFIO2 angegebene Adresse des Ein-/Ausgabebereichs-2 überprüft und dieser Bereich für die anstehende Ein-/Ausgabeoperation verwendet. Ist das Bit nicht gesetzt, wird die im Feld IDMFIO1 enthaltene Adresse des Ein-/Ausgabebereichs-1 überprüft, Ein-/Ausgaben erfolgen dann in diesen bzw. von diesem Bereich aus (siehe Abschnitt „Überlappte Ein-/Ausgabe“ auf Seite 58).

Logische Blocknummer (IDMFLBN)

Die logische Blocknummer ist eine 2 Byte lange Binärzahl ($0 \leq n \leq 65535$). Hat sie den Wert 0, erfolgt die Verarbeitung sequenziell: Beim Lesen wird der Block übertragen, der unmittelbar auf den letzten in einer Lese-/Schreiboperation angesprochenen Block folgt. Beim Schreiben wird ein Block an das Dateiende angefügt.

Ist die logische Blocknummer $\neq 0$, so verweist sie direkt auf den Block der Datei, der gelesen oder geschrieben werden soll.

Bei geketteter Ein-/Ausgabe gelten die Angaben für den ersten Block der Übertragungskette.

Dateiname (IDMFFN)

Der Dateiname ist eine 2 Byte lange Binärzahl (dezimal: $1 \leq n \leq 14000$), die beim Eröffnen einer neuen Datei von EAM in das Feld „Dateiname“ übertragen wird. Diese Zahl ist dort bei jeder späteren Bezugnahme auf die Datei anzugeben.

Fehlerbyte (IDMFEB)

Ist bei einer vom EAM-Makro angestoßenen Operation ein Fehler aufgetreten, werden je nach Ursache entsprechende Bits, die über symbolische Namen ansprechbar sind, gesetzt; gleichzeitig erhält Register 15 den Wert X'00000004'.

IDMFIC	Unzulässige Operation - zum Beispiel unzulässiger Operationsschlüssel, Zugriff auf eine nicht geöffnete Datei, bei geketteter Ein-/Ausgabe liegt der Wert im Feld IDMFNHP nicht zwischen 1 und 16, MFCB ist nicht auf Wortgrenze ausgerichtet usw.
IDMFIF	Unzulässiger Dateiname - Die im Feld „Dateiname“ (IDMFFN) angegebene Zahl bezeichnet keine in diesem Auftrag existente EAM-Datei.
IDMFIB	Ungültige Blocknummer - Die im Feld IDMFLBN (Logische Blocknummer) angegebene Blocknummer liegt außerhalb der Datei (beim Lesen) oder ist größer als die Nummer des zuletzt geschriebenen Blocks + 1 (beim Schreiben).
IDMFIA	unzulässige Adresse eines Ein- oder Ausgabebereichs - Die in den Feldern IDMFIO1 oder IDMFIO2 enthaltene Adresse für Ein-/Ausgabebereich-1/2 ist ungültig.
IDMFNS	Kein EAM-Speicherplatz mehr verfügbar - Z.B.: Der Benutzer hat 14000 EAM-Dateien erzeugt oder der gesamte für EAM-Dateien im System zur Verfügung stehende Speicherplatz ist ausgeschöpft.
IDMFNP	Unzulässiger Zugriff auf eine privilegierte Datei - Ein nichtprivilegierter Benutzer versuchte auf eine privilegierte Datei zuzugreifen.
IDMFEF	Dateiende - Bei einem Lesevorgang wurde das Dateiende erreicht: Überstreicht bei geketteter Eingabe eine Übertragungskette das Dateiende, wird soweit wie möglich ausgelesen und das Bit für Dateiende gesetzt.
IDMFERR	Statusbytes überprüfen - Die vorhergehende Lese- oder Schreiboperation ist nicht erfolgreich abgeschlossen worden. Es sollten die Statusbytes überprüft werden, um die Fehlerursache festzustellen.

Statusfeld (IDMFSB)

Dieses Feld wird vom System versorgt, wenn gleichzeitig folgende Bedingungen erfüllt sind:

– die vorausgehende Operation war eine Lese- oder Schreiboperation;
– die aktuelle Operation ist eine der Operationen Lesen, Schreiben, Prüfen, Prüfen und Warten oder Schließen.

Es werden folgende Bytes aus dem CCB (Channel Control Block) übertragen: das Standard-Gerätebyte, 3 Fehlerbytes, das Ablaufteil-Markierungsbyte.

Adresse von Ein-/Ausgabebereich-1 (IDMFIO1/IDMFIOA1)

Dieses Feld enthält die virtuelle Adresse des ersten Bytes von Ein-/Ausgabebereich-1. Beim Schreiben wird von dieser Adresse ein Block bzw. eine Übertragungskette übertragen, beim Lesen wird der gelesene Block/die Blockfolge an diese Adresse übertragen.

Hat der Ein-/Ausgabebereich Blockgröße (2048 Byte), sollte er innerhalb einer Seite (4096 Byte) liegen und auf Wortgrenze ausgerichtet sein. Bei geketteter Ein-/Ausgabe sollte der Bereich auf Seitengrenze ausgerichtet sein; er muss mindestens so viele Blöcke aufnehmen können, wie mit einer Ein-/Ausgabeanforderung übertragen werden.

Wenn diese Ausrichtungsbedingungen nicht erfüllt werden, kann es bei bestimmter Hardware zu Zwischenpufferung und dadurch zu Performance-Einbuße kommen.

Anzahl zu übertragender Blöcke (IDMFNHP)

Dieses Feld wird zum Zeitpunkt des Eröffnens oder Wiedereröffnens einer Datei ausgewertet, wenn im Zusatzbyte gekettete Ein-/Ausgabe (IDMFCI) verlangt wird. Es enthält eine 1-Byte-Binärzahl ≤ 16.

Bei geketteter Ein-/Ausgabe und Erreichen des Dateiendes im Lesemodus übergibt das System hier die Anzahl der übertragenen Blöcke.

Adresse von Ein-/Ausgabebereich-2 (IDMFIO2/IDMFIOA2)

Dieses Feld enthält die virtuelle Adresse des ersten Bytes von Ein-/Ausgabebereich-2. Die Adresse kann mit der von Ein-/Ausgabebereich-1 übereinstimmen. Soll jedoch zeitlich überlappte Verarbeitung durchgeführt werden, ist hier die Adresse eines Bereichs anzugeben, der sich mit Ein-/Ausgabebereich-1 nicht überschneidet.

Es gelten die Bedingungen für Ein-/Ausgabebereich-1.

Gekettete Ein-/Ausgabe

Ist im MFCB das Bit IDMFCI für gekettetete Verarbeitung gesetzt, ist die Ein-/Ausgabe schneller. Dabei müssen die gekettet zu schreibenden Blöcke nicht nebeneinander liegen. Beim Schreiben werden immer Gruppen von 3 PAM-Seiten gekettet geschrieben, beim Lesen werden nebeneinander liegende Seiten gekettet gelesen. Daher sollte das Feld IDMFNHP (= Anzahl der zu übertragenden Blöcke) ein Vielfaches von 3 enthalten. Außerdem sollten Ein-/Ausgabeoperationen immer bei Blocknummern beginnen, die in der Form (3 * n) + 1 darstellbar sind, d.h. 1, 4, 7, ...

Hinweis

Zur Unterstützung der NK4-Pubsets wird vom EAM-Anwender gefordert, die gekettete Verarbeitung auf den Blockungsfaktor 2 oder ein Vielfaches von 2 umzustellen. In diesem Fall sollten bei direkten Ein-/Ausgabeoperationen ungerade Blocknummern (BLOCK#) angegeben werden, also 1, 3, 5, ...

Durch eine Erhöhung des Blockungsfaktors auf Kosten des Hauptspeicherplatzes (Ein/Ausgabepuffer) werden die CPU-Zeit (Anstoß und Beendigung der Ein- und Ausgabeanforderung) die Kanal- und Gerätezeit (Seek- und Searchzeiten) durch das Lesen bzw. Schreiben von mehreren Blöcken mit einer physikalischen Ein-/Ausgabe) eingespart. Diese Optimierung kann vom System nicht beeinflusst werden, da sie primär in der Verantwortung des EAM-Anwenders liegt.

EAM-Verarbeitung

Verwendung von Prüfoperationen

Nach einem Lese- oder Schreibaufruf erhält der Benutzer die Steuerung zurück, sobald die angeforderte Operation akzeptiert wird. Diese Operation braucht also noch nicht beendet zu sein.

Bevor eine Lese- oder Schreiboperation angestoßen wird, wird die Beendigung einer eventuell vorhergehenden Lese- oder Schreiboperation abgewartet (mit einer impliziten Prüf- und Warteoperation). Ebenso wird die Beendigung der letzten Lese- oder Schreiboperation abgewartet, wenn eine Schließoperation angefordert wurde.

Eine Prüfoperation ist also nur notwendig nach der letzten einer Reihe von Lese- oder Schreiboperationen, wenn die Datei nicht sofort wieder geschlossen wird oder wenn mit geketteter Ein-/Ausgabe bis zum Erreichen der Dateiendebedingung gelesen wird (im Fehlerbyte: IDMFIB oder IDMFEF = 1) und die Anzahl der übertragenen Blöcke ≠ 0 ist.

Beispiel

In einer EAM-Datei werden 3 Leseoperationen durchgeführt. Die Datei wird danach nicht wieder geschlossen, weil sie für spätere Ein-/Ausgaben noch benötigt wird. Diese Ein-/Ausgabeoperationen werden jedoch erst nach der Verarbeitung der gelesenen Blöcke angefordert:

```
LESEN

PRÜFEN/WARTEN              Die Beendigung der letzten Ein-/Ausgabe-
  .                        operation wird abgewartet.
  .
Verarbeitung der gelesenen Blöcke
  .
weitere Ein-/Ausgabeoperationen
```

Änderung von Verarbeitungseigenschaften

Beim Eröffnen oder Wiedereröffnen einer Datei werden folgende Angaben berücksichtigt:

– gekettete/nichtgekettete Ein-/Ausgabe
– Anzahl der zu übertragenden Blöcke

Soll eine dieser Angaben während der Dateiverarbeitung geändert werden, ist folgendes Vorgehen erforderlich:

1. Datei schließen
2. Felder im MFCB modifizieren
3. Datei wieder eröffnen

Die Anzahl der zu übertragenden Blöcke ist z.B. dann zu ändern, wenn mit der letzten Schreiboperation weniger Blöcke übertragen werden sollen, als beim Eröffnen der Datei angegeben wurde.

Beispiel

In eine EAM-Datei sind 99 Blöcke zu schreiben. Es sollen jeweils 15 Blöcke gekettet übertragen werden (Byte IDMFNHP im MFCB). Folgende Operationen werden dann angefordert:

```
ÖFFNEN (neue Datei) -> SCHREIBEN -> SCHREIBEN -> SCHREIBEN -> SCHREIBEN ->
SCHREIBEN -> SCHREIBEN -> SCHLIESSEN *) -> WIEDERERÖFFNEN -> SCHREIBEN ->
SCHLIESSEN
```

*) Nach Ausführung der sechs Schreiboperationen sind noch 9 Blöcke zu schreiben. Deshalb muss die Datei geschlossen und wieder eröffnet werden mit dem Wert 9 für die Anzahl zu übertragender Blöcke.

Überlappte Ein-/Ausgabe

Die Verarbeitungszeit kann durch asynchrone Ein-/Ausgaben verkürzt werden. Nach Anstoßen einer Ein-/Ausgabeoperation wird die Steuerung sofort an das Programm zurückgegeben, wo parallel zur physikalischen Ein-/Ausgabe weitere Verarbeitung stattfindet.
Die nächste Ein-/Ausgabeoperation wird angestoßen für einen zweiten, sich mit dem ersten Bereich nicht überschneidenden Ein-/Ausgabebereich ... usw. Bild 1 auf Seite 59 zeigt die Überlappung von Verarbeitung und Eingabeoperationen.

Sequenzielles Lesen mit geketteter Eingabe

Gekettete Eingabe wird bei sequenziellem Lesen am wirkungsvollsten eingesetzt, wenn

– als Anzahl der zu übertragenden Blöcke ein Vielfaches von 3 gewählt wird (3 n);
– als Blocknummer eine Zahl der Form: (3n) + 1 (z.B. 1, 4, 7 ...) gewählt wird.

Hinweis

Zur Unterstützung der NK4-Pubsets wird vom EAM-Anwender gefordert, die gekettete Verarbeitung auf den Blockungsfaktor 2 oder ein Vielfaches von 2 umzustellen. In diesem Fall sollten bei direkten Ein-/Ausgabeoperationen ungerade Blocknummern (BLOCK#) angegeben werden, also 1, 3, 5, ...

Das so umgestellte Programm ist abwärtskompatibel und damit auch ablauffähig in Vorgängerversionen.

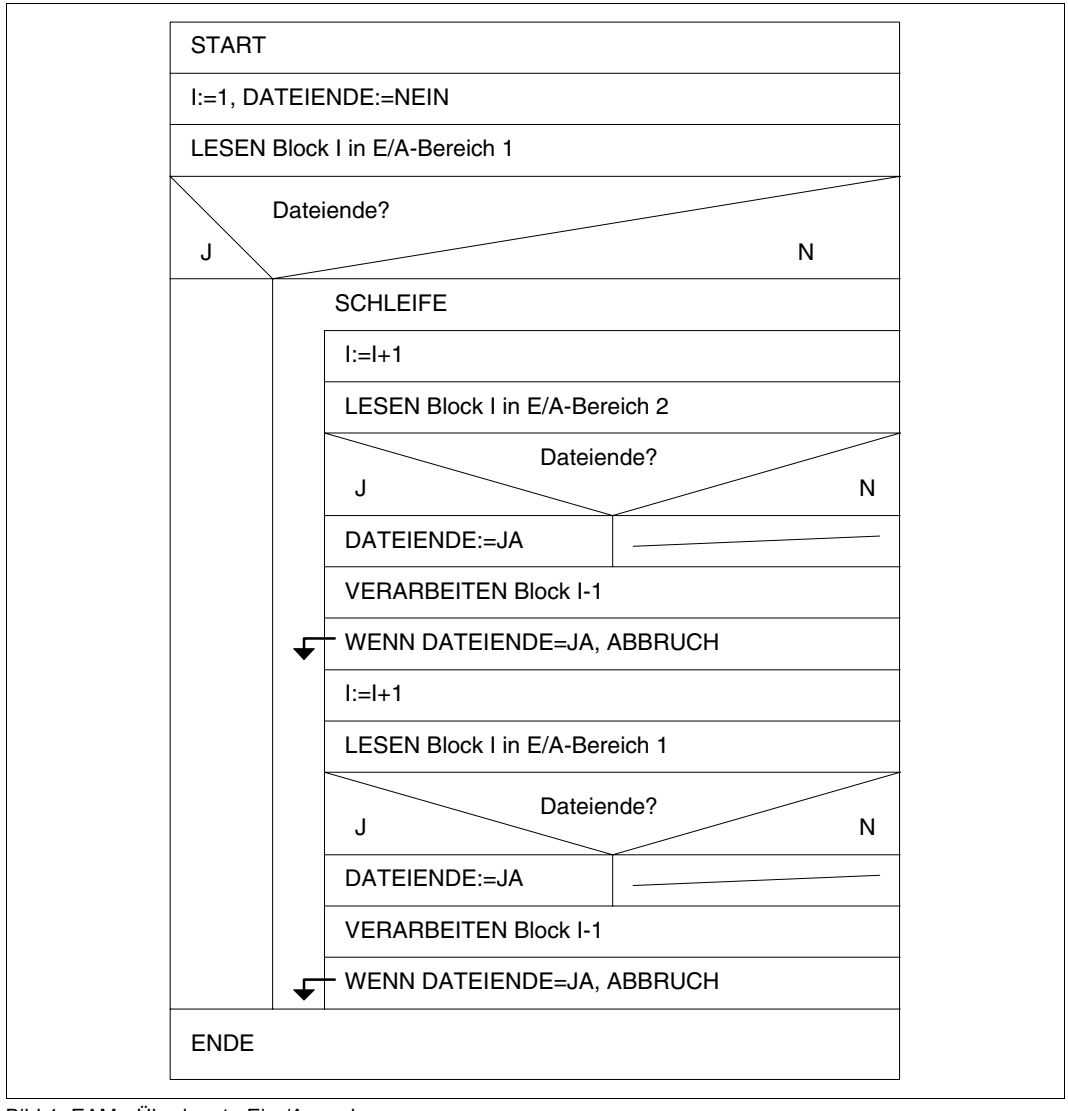


Bild 1: EAM - Überlappte Ein-/Ausgabe

Behandlung von Bindemoduldateien durch EAM

Jeder Auftrag kann genau eine Bindemoduldatei bearbeiten. Ist das Bit IDMFOO des Zusatzbytes gesetzt, beziehen sich alle Operationen auf die Bindemoduldatei. Folgendes Diagramm zeigt die Aktionen beim Eröffnen oder Wiedereröffnen einer Datei:

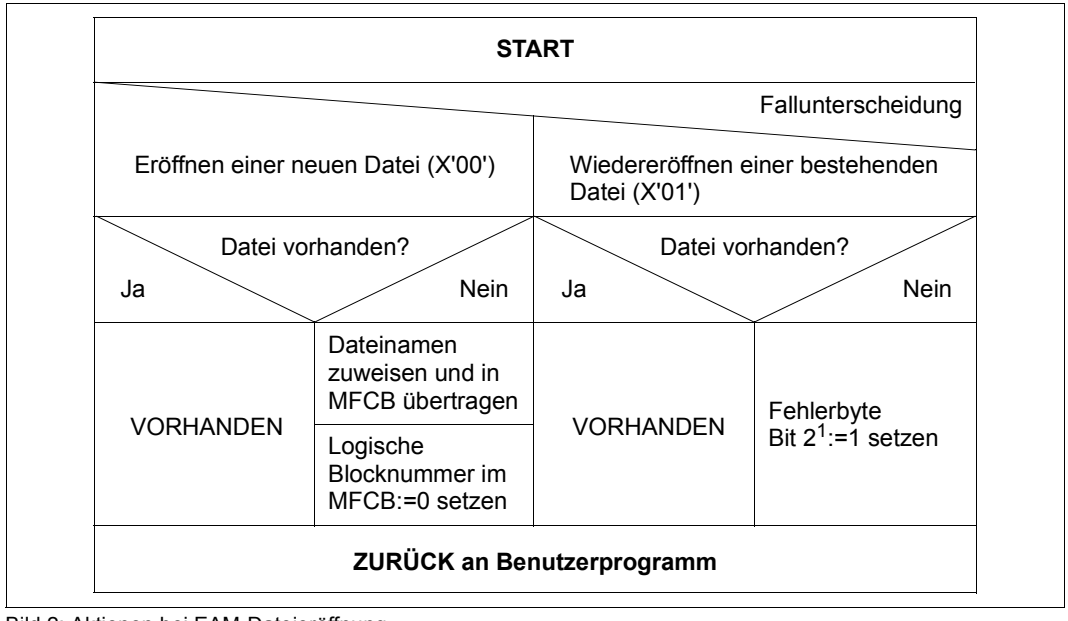


Bild 2: Aktionen bei EAM-Dateieröffnung

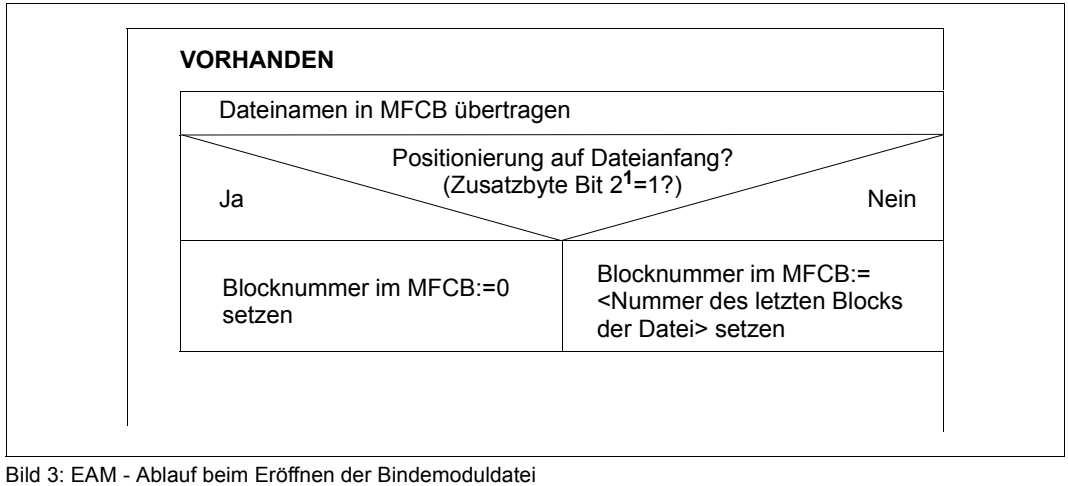


Bild 3: EAM - Ablauf beim Eröffnen der Bindemoduldatei

3.4 FASTPAM - Fast Primary Access Method

FASTPAM (Fast Primary Access Method) ist eine Blockzugriffsmethode für NK4-Plattendateien. Sie bietet eine mit UPAM vergleichbare Funktionalität, wobei jedoch die Performance gegenüber UPAM erheblich verbessert wurde und Mehrrechnersysteme besonders berücksichtigt werden.

Mit FASTPAM ist es möglich, Ein-/Ausgaben direkt in Datenräumen (Data Spaces) durchzuführen. Dazu werden IO-Area-Pools in Datenräume gelegt; allerdings lassen sich diese IO-Area-Pools nur nicht-resident anlegen.

Die Zugriffsmethode FASTPAM arbeitet mit einem speziellen OPEN.

Bei der FASTPAM-Schittstelle handelt es sich um eine SVC-Schnittstelle. Die Aufträge werden durch eine Parameterliste formuliert; Rückmeldungen über das Ergebnis erfolgen über einen Returncode in der Parameterliste (nicht über Exits).

Makroaufrufe für die Zugriffsmethode FASTPAM

Die Zugriffsmethode FASTPAM arbeitet mit den beiden folgenden Makros:

Makro	Funktion
FPAMSRV	Verwaltungsfunktionen: – Systemumgebung (FASTPAM-Environment) vorbereiten – Ein-/Ausgabebereiche (FASTPAM-IO-Area-Pool) vorbereiten – Datei zur Bearbeitung eröffnen – eine mit FPAMSRV geöffnete Datei schließen – Systemumgebung (FASTPAM-Environment) abbauen – Ein-/Ausgabebereiche (FASTPAM-IO-Area-Pool) abbauen
FPAMACC	Zugriffsfunktionen (Dateizugriffe formulieren): – synchrones Lesen und Schreiben von logischen Blöcken – asynchrones Lesen und Schreiben von logischen Blöcken – Warten auf die Beendigung asynchroner Ein-/Ausgabeaufträge – Benachrichtigung über die Beendigung asynchroner Ein-/Ausgabeaufträge

FASTPAM-Funktionen

Die Funktionen der Zugriffsmethode FASTPAM sind in den beiden Makros FPAMSRV und FPAMACC realisiert. Es handelt sich dabei um folgende Funktionen:

Funktion	Bedeutung
ENABLE ENVIRONMENT	Systemumgebung für FASTPAM-Bearbeitung vorbereiten
ENABLE IOAREA POOL	IO-Area-Bereich für FASTPAM-Bearbeitung vorbereiten
OPEN FILE	Datei zur Bearbeitung mit FASTPAM öffnen
ACCESS FILE	(eine mit FPAMSRV geöffnete) Datei bearbeiten
CLOSE FILE	(eine mit FASTPAM geöffnete) Datei schließen; hierbei kann auf Wunsch der Last Page Pointer angegeben werden.
DISABLE ENVIRONMENT	Systemumgebung für FASTPAM-Bearbeitung abbauen
DISABLE IOAREA POOL	IO-Area-Bereich für FASTPAM-Bearbeitung abbauen

*Systemumgebung für FASTPAM-Bearbeitung vorbereiten (Makrofunktion FCT=*ENAENV)*

Mit der Funktion ENABLE ENVIRONMENT (Makro FPAMSRV, Operand FCT=*ENAENV) kann ein Anwender ein FASTPAM-Environment einrichten oder seine Teilnahme an einem existierenden FASTPAM-Environment erklären. Dem Aufrufer wird vom System eine taskspezifische Environment Kurzkennung zurückgegeben, mit der er sich bei den nachfolgenden OPEN-Aufrufen auf das Environment beziehen kann.

Da ein FASTPAM-Environment über seinen Namen und seinen Geltungsbereich identifiziert wird, und der Geltungsbereich implizit aus der Adresse der FPAMACC-Parameterlisten abgeleitet wird, muss bei jedem ENABLE-ENVIRONMENT-Aufruf sowohl der Name als auch die Adresse der Parameterlisten angegeben werden. Die anderen Attribute brauchen beim Anschluss an ein bestehendes Environment nicht erneut angegeben zu werden. Werden sie dennoch spezifiziert, so müssen die angegebenen Werte mit denjenigen des bestehenden Environments übereinstimmen.

Verfügt der Anwender über die FASTPAM-Berechtigung, wird der gesamte für die Plattenzugriffe notwendige Klasse-3-Speicher vorgeneriert und der Bereich der FPAMACC-Parameterlisten fixiert. Damit dies durchgeführt werden kann, wird die Adresse des Parameterlistenbereichs und die Anzahl der Parameterlisten, sowie die bei den späteren Dateizugriffen benutzte maximale Übertragungslänge benötigt.

Für die Übertragungslänge sind nur die Werte 4 KByte und 32 KByte zugelassen. Der Wert 32 KByte sollte nur bei nicht zu hoher Zugriffsparallelität verwendet werden, da pro Ein-/Ausgabepfad 2 KByte residenter Systemspeicher belegt wird. Die logische Blocklänge der Dateien, die später mit diesem Environment geöffnet werden, und die Ein-/Ausgabelänge der folgenden Dateizugriffe darf diesen Maximalwert nicht überschreiten.

Will der Anwender mit Eventing arbeiten, muss er die Kurzkennung der Ereigniskennung schon beim Einrichten des Environment angeben.

Der Parameterlistenbereich muss vorher angefordert worden sein und schreibenden Zugriff erlauben. Während des Einrichtens des FASTPAM-Environments durch den ersten Environment-Teilnehmer (während der Bearbeitung des ENABLE-ENVIRONMENTS-Aufrufs), darf auf keiner Seite, die sich mit dem Parameterlistenbereich überlappt, eine Ein-/Ausgabe laufen.

*IO-Area-Bereich für FASTPAM-Bearbeitung vorbereiten (Makrofunktion FCT=*ENAIPO)*

Mit der Funktion ENABLE IOAREA POOL (Makro FPAMSRV, Operand FCT=*ENAIPO) kann ein Anwender einen FASTPAM-IO-Area-Pool einrichten oder seine Teilnahme an einem existierenden FASTPAM-IO-Area-Pool erklären. Das Betriebssystem fixiert bei vorhandener FASTPAM-Berechtigung den angegebenen Speicher und gibt dem Aufrufer eine taskspezifische IO-Area-Pool-Kurzkennung zurück, mit der er sich bei den nachfolgenden OPEN-FASTPAM-Aufrufen auf den Pool beziehen kann.

Analog zum FASTPAM-Environment wird ein IO-Area-Pool über seinen Namen und seinen Geltungsbereich eindeutig identifiziert. Beim Einrichten des IO-Area-Pools werden die Attribute unveränderlich festgelegt. Jeder weitere Teilnehmer darf keine abweichenden Attribute spezifizieren. Dies kann er z.B. durch alleinige Angabe des Namens und der Adresse erreichen.

Der Speicherbereich muss vorher angefordert worden sein und schreibenden Zugriff erlauben. Während des Einrichtens des FASTPAM-IO-Area-Pools durch den ersten IO-Area-Teilnehmer (während der Bearbeitung des ENABLE-IOAREA-POOL-Aufrufs), darf auf keiner Seite des IO-Area-Pools eine Ein-/Ausgabe laufen.

*Datei zur Bearbeitung mit FASTPAM öffnen (Makrofunktion FCT=*OPEN)*

Mit den beim ENABLE ENVIRONMENT und ENABLE IOAREA POOL erhaltenen Kurzkennungen kann jede Task beliebig viele Dateien eröffnen.
Ist an das FASTPAM-Environment eine Ereigniskennung gekoppelt, so besteht die Möglichkeit, jede Datei mit FPAMSRV, Operand FCT=*OPEN, EVENTNG=*YES zu eröffnen. Es wird dann jeder asynchrone ACCESS-Auftrag über die Ereigniskennung quittiert. Ansonsten ist jeder asynchrone Auftrag vom Anwender explizit mit einer Wait-Operation zu beenden. Synchrone Aufträge werden in beiden Fällen gleich behandelt.
Mit dem Operanden MODE kann die Zugriffsart (lesend oder schreibend) und mit dem Operanden SHARUPD die Multi-User-Betriebsart definiert werden. Der Operand BLKSIZE bestimmt das Granulat der folgenden Dateizugriffe.
Mit dem Operanden LARGE_FILE wird angegeben, ob die zu eröffnende Datei eine „große Datei“ werden darf, die eine Dateigröße ≥ 32 GB erlaubt.
Für jeden fehlerfrei beendeten OPEN erhält der Aufrufer eine OPEN-Kurzkennung, die bei den folgenden ACCESS-FILE-Aufträgen anzugeben ist.

Analog zu UPAM dominieren die bei einem vorher abgesetzten ADD-FILE-LINK-Aufruf angegebenen Parameterwerte über die beim *OPEN angegebenen Werte. Sind sie beim FASTPAM-OPEN nicht erlaubt, so wird der *OPEN-Aufruf zurückgewiesen.

Eine mit FPAMSRV geöffnete Datei bearbeiten (Makro FPAMACC)

Mit dem Makro FPAMACC kann auf die Datei schreibend oder lesend zugegriffen werden. Die Datei, das zugehörige Environment und der IO-Area-Pool werden dabei mittels OPEN-Kurzkennung spezifiziert. Die Ein-/Ausgaben können synchron und asynchron in Auftrag gegeben werden.

Synchrone Operationen sind:

– READ AND WAIT
– WRITE AND WAIT
– READ AND EQUALIZE

Asynchrone Operationen sind:

– READ
– WRITE

Mit der WAIT-Operation wird das Ende asynchroner Aufträge, die nicht synchron durchgeführt wurden, abgewartet.

Um dem Anwender eine performante Bedienung von Dateien zu ermöglichen, die in einem Zwischenpuffer (Cache) gelagert sind, schließt FASTPAM synchron ausgeführte asynchrone Aufträge vollständig ab und sendet bei Teilnahme am Eventing kein Signal an die Ereigniskennung. Nach einem synchron ausgeführten Auftrag darf kein WAIT-Makro abgesetzt werden (bzw. bei Eventing darf kein SOLSIG-Aufruf abgesetzt werden).

Die Ein-/Ausgabelänge muss ein Vielfaches von BLKSIZE sein und darf den bei MAXIOLN angegebenen Wert nicht überschreiten. Notwendige Angaben sind ferner der logische Block innerhalb der Datei (BLOCK) und die Adresse des Ein-/Ausgabepuffers (IOAREA).

Um SVCs einzusparen, wird auch die Auftragskettung unterstützt. Mit dem Operanden CHAIN können bis zu 5000 FPAMACC-Listen miteinander verbunden werden.

Eventing Verarbeitung

FASTPAM bietet analog zu UPAM die ereignisgesteuerte Verarbeitung von Ein-/Ausgaben an (siehe Abschnitt „TU-Eventing - ereignisgesteuerte Verarbeitung“ auf Seite 106 und Handbuch „Makroaufrufe an den Ablaufteil“ [2]). Wurde der Auftrag nicht synchron beendet, gibt FASTPAM nach Vollendung einer Ein-/Ausgabe eine Meldung an die zugeordnete Ereigniskennung, die vom Anwender mit dem Makroaufruf SOLSIG abgeholt wird. Bei Angabe von EVENTING=YES erfolgt eine Benachrichtigung; bei Angabe von EVENTING=NO erfolgt keine Benachrichtigung. Es muss ein WAIT abgesetzt werden.

*Eine mit FASTPAM geöffnete Datei schließen (Makrofunktion FCT=*CLOSE)*

Mit der Funktion CLOSE (Makro FPAMSRV, Operand FCT=*CLOSE) wird eine geöffnete Datei wieder geschlossen. Auch hier wird die Datei mittels OPEN-Kurzkennung identifiziert.

*Systemumgebung für FASTPAM-Bearbeitung abbauen (Makrofunktion FCT=*DISENV)*

Mit der Funktion DISABLE ENVIRONMENT (Makro FPAMSRV, Operand FCT=*DISENV) löst eine Task die Verbindung zu einem mittels Kurzkennung spezifizierten FASTPAM-Environment. Bei der Diskonnektierung der letzten Task wird das FASTPAM-Environment abgebaut. Der Anwenderspeicher wird dabei nicht freigegeben.

*IO-Area für FASTPAM-Bearbeitung abbauen (Makrofunktion FCT=*DISIPO)*

Mit der Funktion DISABLE IOAREA POOL (Makro FPAMSRV, Operand FCT=*DISIPO) löst eine Task die Verbindung zu einem mittels Kurzkennung spezifizierten FASTPAM-IO-Area-Pool. Bei der Diskonnektierung der letzten Task wird der IO-Area-Pool abgebaut.
Der Anwenderspeicher wird dabei nicht freigegeben.

Dateiverarbeitung mit FASTPAM

Dateiformat

FASTPAM bearbeitetet nur PAM-Dateien mit BLKCTRL=NO/DATA und BLKSIZE=(STD,2n), wobei n=1,2,3...8 ist. Dateien, die dieses Format nicht aufweisen, müssen zunächst konvertiert werden.

FASTPAM-Berechtigung

Der Benutzer benötigt einen Eintrag im Benutzerkatalog (Kommando SHOW-USER-ATTRIBUTES, Feld `DMS-TUNING-RESOURCES=*EXCLUSIVE` bzw. Kommando MODIFY-USER-ATTRIBUTES, Feld `DMS-TUNING-RESOURCES=*EXCLUSIVE-USE`), der ihn berechtigt, über FASTPAM-Aufrufe residenten Speicher zu erhalten. Hat er diese Berechtigung nicht, so kann der Benutzer zwar mit der Zugriffsmethode FASTPAM arbeiten, doch werden keine Bereiche resident gehalten. FASTPAM verhält sich in diesem Fall wie UPAM: vom System wird nur ein kleiner, nicht-residenter Teil der Ein-/Ausgabepfade angelegt; der Bereich der Parameterlisten und des IO-Area-Pool wird nicht fixiert. Die Folge ist, dass bei jeder Ein-/Ausgabe die Pfade neu angelegt und die Anwenderbereiche validiert und fixiert werden müssen, wodurch die FASTPAM-typischen Performance-Gewinne verloren gehen.

Für den Fall, dass keine Speicherresidenz erreicht werden kann, verhält sich FASTPAM wie bei Fehlen der FASTPAM-Berechtigung. Es wird dadurch ein dem UPAM äquivalentes oder besseres Performance-Verhalten geboten.

Speicherbereiche resident machen

Ein wesentlicher Zweck von FASTPAM besteht darin, hochperformante Dateizugriffe zu ermöglichen, indem bereits vor dem ersten Dateizugriff die notwendige Systemumgebung resident bereitgestellt wird.

Hierzu wird der Speicherbereich, der die Anwenderparameterlisten enthält, sowie der Speicherbereich, der die IO-Areas enthält (beide vom Anwender zur Verfügung gestellt), durch das „FASTPAM-Seitenfixieren“ speicherresident gemacht.

Dies ist im Wesentlichen derselbe Vorgang, der bei anderen Zugriffsmethoden bei jeder Ein-/Ausgabe für die IO-Area von PPAM ausgeführt wird. Nur wird bei diesen anderen Zugriffsmethoden die IO-Area nach Abschluss der Ein-/Ausgabe wieder freigegeben.

Bei FASTPAM bestimmt der Anwender durch ENABLE/DISABLE ENVIRONMENT, wie lange die Parameterlisten fixiert sind, und durch ENABLE/DISABLE IOAREA POOL, wie lange die IO-Areas fixiert sind. Während dieser Zeit kann er damit arbeiten. Die Validierung muss nur anfangs einmal erfolgen, da ein Freigeben fixierter Bereiche nicht möglich ist.

Außerdem wird bei ENABLE ENVIRONMENT der für die Ein-/Ausgaben benötigte Systemspeicher angefordert (je 1* pro parallel mögliche IO). Der größte Teil dieses Speichers, der von IOCTRL verwendete Bereich, ist immer resident. Dies ist auch bei anderen Zugriffsmethoden der Fall, doch wird er bei anderen Zugriffsmethoden für jede Ein-/Ausgabe neu zugeteilt und nicht permanent belegt.

Des Weiteren besteht dieser Systemspeicher aus einer FASTPAM-Workarea, die vor allem die Parameterliste zum Aufruf von PPAM enthält; diese muss, im Unterschied zu UPAM, beim Arbeiten mit TU-Eventing ebenfalls im residenten Speicher liegen (da die IO dann in SIH terminiert wird).

Alle diese Fixierungen erfolgen jedoch nur, wenn die Benutzerkennung über die FASTPAM-Berechtigung verfügt (Benutzerkatalog-Eintrag, Feld `DMS-TUNING-RESOURCES=*EXCLUSIVE`).

In diesem Fall - und wenn die betreffenden Fixierungen durchgeführt werden konnten - kann man von einem „residenten“ Environment bzw. einem „residenten“ IO-Area-Pool sprechen.

„Residentes Environment“ bedeutet:
- residente vorvalidierte Parameterlisten
- vorbestellter Systemspeicher

falls mit Eventing gearbeitet wird:
- residente FASTPAM-Workarea

„Residenter IO-Area-Pool„ bedeutet:
- residente vorvalidierte IO-Areas

Voraussetzungen für residente FASTPAM-Bereiche

- der Anwender gab die entsprechenden Parameter an (Makro FPAMSRV, FCT=*ENAENV/*ENAIPO, Operand RES=YES)
- der Anwender verfügt über die FASTPAM-Berechtigung
- es wird nicht mit Datenräumen gearbeitet
- es ist genügend freier Hauptspeicherplatz vorhanden
- die beim Programmaufruf zugeteilten residenten Seiten reichen aus (Kommando START-PROGRAM/LOAD-PROGRAM, Operand RESIDENT-PAGES); Voraussetzung für die Zuteilung residenter Seiten beim Programmaufruf ist, dass die im Benutzerkatalog festgelegte Höchstgrenze und die systemglobale Grenze für residente Speicherseiten nicht überschritten wird.

FASTPAM-Makros und ihre Funktionen

Zur Dateibearbeitung stehen dem Anwender die beiden Makros FPAMSRV und FPAMACC zur Verfügung, mit denen die entsprechenden Funktionen und Operationen ausgeführt werden können (siehe auch Abschnitt FASTPAM-Funktionen, Seite 62, Makro- und Operandenbeschreibung, Seite 549).

Makro FPAMSRV mit den Funktionen:

ENABLE ENVIRONMENT	Systemumgebung für FASTPAM-Bearbeitung vorbereiten
ENABLE IOAREA POOL	IO-Area-Bereich für FASTPAM-Bearbeitung vorbereiten
OPEN FILE	Datei zur Bearbeitung mit FASTPAM öffnen
ACCESS FILE	(eine mit FPAMSRV geöffnete) Datei bearbeiten
CLOSE FILE	(eine mit FASTPAM geöffnete) Datei schließen; hierbei kann auf Wunsch der Last Page Pointer angegeben werden.
DISABLE ENVIRONMENT	Systemumgebung für FASTPAM-Bearbeitung abbauen
DISABLE IOAREA POOL	IO-Area-Bereich für FASTPAM-Bearbeitung abbauen

Makro FPAMACC mit der Funktion:

ACCESS FILE	(eine mit FPAMSRV geöffnete) Datei bearbeiten

Multi-User-Betrieb an einem Rechner

Eine PAM-Datei kann mit den Zugriffsmethoden UPAM (siehe Seite 89), FASTPAM oder DIV (siehe Seite 39) erstellt bzw. bearbeitet werden.

Der erste Anwender (User A) kann beim Eröffnen seiner PAM-Datei eine beliebige Kombination von OPEN-Modus und SHARUPD-Angaben (im FCB-Makroaufruf) wählen.
Die Tabelle auf Seite 69 zeigt, mit welchen Kombinationen von OPEN und SHARUPD ein weiterer Anwender (User B) die bereits eröffnete Datei seinerseits eröffnen kann.
Ist die Datei von mehr als zwei Anwendern eröffnet worden, wird die Kombination OPEN/SHARUPD jedes neu hinzukommenden Anwenders (User B) mit allen bestehenden Eröffnungen (User A) verglichen. Weitere Anwender können die Datei nur eröffnen, wenn jeder dieser Vergleiche positiv ausgefallen ist. Nicht erlaubte Kombinationen führen zu einem OPEN-Fehler.

Für die Zugriffsmethode FASTPAM gilt Folgendes:

– Mehrere parallele Prozesse (mehrere SHARUPD=*YES und MODE=*OUTIN/*INOUT-Eröffner) können eine Datei gleichzeitig mit FASTPAM bearbeiten.

  Hinweis

  Bei Mehrfachzugriff auf die Datei (Shared Update Mode) muss der Anwender selbst für entsprechende Synchronisierungsroutinen sorgen, falls diese vom eingesetzten Softwareprodukt nicht vorgesehen sind. Im Unterschied zu UPAM stellt FASTPAM hierfür keine Blocksperren-Mechanismen zur Verfügung.

– FASTPAM- und UPAM-Eröffner

  Eine Datei kann parallel von mehreren Tasks sowohl mit FASTPAM als auch mit UPAM eröffnet werden. Die Bearbeitung wird dabei von den Operanden MODE und SHARUPD (s.u.) der Funktion OPEN gesteuert. FASTPAM unterstützt zwar SHARUPD=WEAK nicht, verhält sich aber ansonsten genauso wie UPAM, sowohl bei FASTPAM-Eröffnern unter sich, als auch bei gemischten UPAM- und FASTPAM-Eröffnern.

  Wird auf eine Datei gleichzeitig mit UPAM und FASTPAM zugegriffen, so muss auch der UPAM-Anwender sich selbst mit dem FASTPAM-Anwender synchronisieren, da die UPAM-Blocksperren nur bei beidseitiger Anwendung wirken und FASTPAM keinen Blocksperren-Mechanismus zur Verfügung stellt.

– FASTPAM- und DIV-Eröffner

  FASTPAM verhält sich zu DIV genauso wie UPAM. Eine parallele Verarbeitung ist nur erlaubt, wenn die Datei von allen mit INPUT eröffnet wird.

Verträglichkeits-Matrix FASTPAM mit UPAM/FASTPAM/DIV

FASTPAM unterstützt nicht SHARUPD=*WEAK

			USER B								
			SHARUPD =								
			*YES			*NO			*WEAK		
OPEN-Modus			INPUT	INOUT	OUTIN	INPUT	INOUT	OUTIN	INPUT	INOUT	OUTIN
USER A	SHARUPD =*YES	INPUT INOUT OUTIN	X O O	O O O		X			X X X		
	SHARUPD =*NO	INPUT INOUT OUTIN	X			X			X X X		
	SHARUPD =*WEAK	INPUT INOUT OUTIN	X	X		X	X		X X X	X	

X: OPEN erlaubt

O: OPEN nur erlaubt, wenn die Eröffner dieselbe blockorientierte Zugriffsmethode benutzen (nur UPAM/FASTPAM oder nur DIV)

und denselben Wert für den Operanden LOCKENV benutzen (alle LOCKENV=*HOST oder LOCKENV=*XCS)

und alle im selben HOST laufen *oder* in einem XCS-Verbund bei Verwendung von LOCKENV=*XCS

Anmerkungen

– Leseoperationen mit SHARUPD=*WEAK können eine Datei gleichzeitig mit jeder beliebigen Schreiboperation eröffnet haben. (SHARUPD=*WEAK ist nur bei UPAM und DIV möglich.)
Ausnahme: Für Leseoperationen mit DIV-SHARUPD=*WEAK, die bei OPEN LOCVIEW=*MAP angegeben haben, ist eine parallele Eröffnung mit einer UPAM-/FASTPAM-Schreiboperation nicht erlaubt.

  Für Leseoperationen mit DIV-SHARUPD=*WEAK, die bei OPEN LOCVIEW=*NONE spezifiziert haben, besitzen dieselbe Verträglichkeit wie Leseoperationen mit UPAM-/FASTPAM-SHARUPD=*WEAK.

– Eröffner mit DIV-SHARUPD=*YES sind nicht mit Eröffnern mit UPAM-/FASTPAM-SHARUPD=*YES verträglich.

– Leseoperationen sind immer miteinander verträglich (unabhängig von Zugriffsmethode, SHARUPD-Spezifikation, LOCKENV-Spezifikation und Host.

– Nicht erlaubte Kombinationen führen zu einem OPEN-Fehler.

– SHARUPD=*YES:
Bei jedem Aufruf des Allocators wird die Dateigröße überprüft.
Wenn bei dieser Überprüfung eine Dateigröße ≥ 32 GB ermittelt wird und im zugehörigen FCB das Attribut LARGE_FILE=*FORBIDDEN bzw. in der TFT das Attribut EXCEED-32GB=*FORBIDDEN gesetzt ist, wird die Verarbeitung abgebrochen.
FASTPAM liefert in diesem Fall den Returncode `X'00400145'` in seiner eigenen Parameterliste FPAMACC(I).

Multi-User-Betrieb an mehreren Rechnern

Multi-System-Umgebung ist der Verbund mehrerer Systeme untereinander mittels mehrbenutzbarer privater Platten (siehe Kapitel „Datenträger“, Handbuch „Einführung in das DVS“ [1]) oder Shared-Pubsets. Die Kombinationen für den Zugriff von zwei Systemen aus sind in der Tabelle auf Seite 69 dargestellt.

Für die Zugriffsmethode FASTPAM gilt Folgendes: Multi-System-Umgebung ist der Verbund mehrerer Systeme untereinander mittels Shared-Pubsets.

– Dateien auf Shared-Pubsets werden von FASTPAM unterstützt: FASTPAM-Eröffner können von verschiedenen Rechnern auf einen Shared-Pubset lesend zugreifen, auch parallel mit UPAM- und DIV-Eröffnern, die lesend zugreifen.
– Dateien auf Shared-Private-Disk (SPD) und Protected-Private-Disk (PPD) werden von FASTPAM nicht unterstützt.
– Rechnerübergreifender Dateizugriff (RFA) wird nicht unterstützt.

Datenkonsistenz

– Datenkonsistenz im Multi-User-Betrieb

  Die Zugriffsmethode FASTPAM bietet bei Mehrfachzugriff auf eine Datei (Shared-Update-Mode) **keine** Synchronisierungsmechanismen an. Der Anwender muss daher **selbst** für entsprechende Synchronisierungsroutinen sorgen, falls diese vom eingesetzten Softwareprodukt nicht vorgesehen sind. Beim gemeinsamen Shared-Update-Betrieb einer FASTPAM-, UPAM- und ggf. DIV-Anwendung muss für alle Zugriffe ein gemeinsamer Synchronisierungsmechanismus verwenden werden.

– Datenkonsistenz nach einem Systemausfall

  Tritt bei einem ACCESS-FILE-Auftrag ein Fehler auf, kann nicht angegeben werden, ob und wie viele Daten übertragen worden sind. Das Schreiben eines Blocks darf nicht als atomar angenommen werden. Der Inhalt der Datei kann in einem solchen Fall in einem inkonsistenten Zustand sein.

Zusammenfassung der funktionellen Unterschiede zwischen UPAM und FASTPAM

- Mit FASTPAM können ausschließlich PAM-Dateien mit folgenden Dateieigenschaften verarbeitet werden:
  - BLOCK-CONTROL-INFO=NO
  - BUF-LEN geradzahlig
- Mit FASTPAM können Ein-/Ausgaben in Datenräume (Data Spaces) erfolgen.
- Folgende Funktionen werden von FASTPAM nicht unterstützt:
  - DUMMY-Dateien
  - Bandverarbeitung
  - RFA
  - SPD (Shared Private Disc)
  - PPD (Protected Private Disc)
- FASTPAM unterstützt synchrone und asynchrone Lese- und Schreiboperationen. Folgende von UPAM angebotene Operationen werden von FASTPAM nicht unterstützt:
  - CHK
  - LOCK / UNLOCK
  - LRD / LRDWT / WRTWU
  - SETL
  - SYNC
- Die Funktionalität der UPAM-Operation SETLPP wird im Rahmen der FASTPAM-CLOSE-Verarbeitung bereitgestellt.
- Die Funktion SHARUPD=*WEAK wird nicht unterstützt (siehe auch „Verträglichkeits-Matrix FASTPAM mit UPAM/FASTPAM/DIV“ auf Seite 69).
- Ein impliziter WAIT ist nicht möglich.
- Innerhalb einer OPEN/CLOSE-Klammer können asynchrone Ein-/Ausgaben entweder **nur** durch WAIT abgeschlossen oder ihre Beendigung **nur** über den Eventing-Mechanismus gemeldet werden.
- Eine relative Seitennummern-Angabe ist nicht möglich.

3.5 ISAM - Indexed Sequential Access Method

ISAM ist wie SAM eine satzorientierte Zugriffsmethode für Plattendateien. Grundlage der Verarbeitung ist eine aus Index- und Datenblöcken aufgebaute Datei. Jeder Datensatz enthält einen Schlüssel; die Schlüssel dienen als Sortierkriterium (Zu Index- und Datenblöcken siehe Handbuch „Einführung in das DVS“ [1]).

Es gibt zwei Ausprägungen der Zugriffsmethode ISAM, mit denen Dateien unterschiedlicher Blockformate (siehe Kapitel „Zugriffsmethoden“ im Handbuch „Einführung in das DVS“ [1]) verarbeitet werden können:

– K-ISAM (Key-ISAM) verarbeitet Dateien des Blockformates „PAMKEY“: Diese Dateien sind dadurch gekennzeichnet, dass für jede PAM-Seite DVS-Verwaltungsinformation in einem eigenen (außerhalb der Seite gelegenen) PAM-Schlüssel geführt wird.

– NK-ISAM (Nonkey-ISAM) verarbeitet Dateien des Blockformats „DATA“: Solche Dateien enthalten keine gesonderten PAM-Schlüssel. Die DVS-Verwaltungsinformation wird innerhalb der PAM-Seite in einem Blockkontrollfeld hinterlegt.

Mit dem Operanden BLKCTRL in den Makros FILE und FCB kann der Anwender zwischen diesen beiden Verarbeitungsformen wählen: BLKCTRL=PAMKEY vereinbart K-ISAM, BLKCTRL=DATA/DATA2K/DATA4K legt NK-ISAM fest.

Makroaufrufe für die Zugriffsmethode ISAM

Die Makros der Zugriffsmethode ISAM lassen sich in Funktionsklassen einteilen:

Verwaltung: Makros mit Verwaltungsfunktionen, die die Dateibearbeitung unterstützen.

Zugriff: Makros, die auf die Daten der Datei zugreifen.

Makro-aufruf	Funktions-klasse	Kurzbeschreibung
ADDPLNK	Verwaltung	weist einem Benutzer-ISAM-Pool einen Poolkettungsnamen zu
CREAIX	Verwaltung	erzeugt einen Sekundärindex für eine ISAM-Datei
DELAIX	Verwaltung	löscht Sekundärindizes einer ISAM-Datei
DELPOOL	Verwaltung	löscht einen Benutzer-ISAM-Pool
ELIM	Zugriff	streicht einen Satz aus der Datei
GET	Zugriff	liest den in der Datei folgenden Satz (sequenzielles Lesen)
GETFL	Zugriff	bei Verwendung von markierten ISAM-Schlüsseln: liest den folgenden Satz innerhalb des Markierungsbereichs (sequenziell)
GETKY	Zugriff	liest den ersten Satz mit dem angegebenen Schlüssel
GETR	Zugriff	liest den vorhergehenden Satz (sequenzielles Lesen, rückwärts)
INSRT	Zugriff	fügt einen Satz mit neuem ISAM-Schlüssel in die Datei ein
ISREQ	Zugriff	hebt eine ISAM-Sperre auf
OSTAT	Zugriff	informiert über Anzahl und Art gleichzeitiger Dateizugriffe
PUT	Zugriff	schreibt sequenziell Sätze an das Dateiende (inklusive Prüfung der Schlüssel auf gültige Reihenfolge)
PUTX	Zugriff	ersetzt einen durch GET o.Ä. bereitgestellten Satz
REMPLNK	Verwaltung	löscht den Poolkettungsnamen
RETRY	Zugriff	setzt nach Durchlaufen des EXLST-PGLOCK-Ausgangs den ISAM-Zeiger neu und wiederholt den letzten Makroaufruf
SETL	Zugriff	positioniert den ISAM-Zeiger für anschließende sequenzielle Verarbeitung auf Dateianfang, -ende oder einen bestimmten Satz
SHOPLNK	Verwaltung	informiert über die Zuordnung von ISAM-Pools zu Poolkettungsnamen
SHOPOOL	Verwaltung	informiert über Attribute und Belegungszustände von ISAM-Pools
STORE	Zugriff	– fügt einen Satz mit neuem ISAM-Schlüssel in die Datei ein – überschreibt einen Satz mit bereits vorhandenem ISAM-Schlüssel, wenn Mehrfachvergabe von ISAM-Schlüsseln nicht zulässig – fügt einen Satz mit bereits vorhandenem ISAM-Schlüssel als letzten Satz mit diesem Schlüssel in die Datei ein

OPEN-Modi

Bevor eine Datei verarbeitet werden kann, muss sie mit einem OPEN-Makroaufruf eröffnet werden. Für ISAM-Dateien sind folgende Eröffnungsmodi zulässig: OUTPUT, OUTIN, EXTEND, INOUT, INPUT. Gleichzeitig muss überprüft werden, ob die Datei bereits von einem anderen Auftrag geöffnet ist, in welchem ISAM-Pool sie verarbeitet werden soll, ob sie im Übertragungs- oder im Locate-Mode zu verarbeiten ist, usw.

OUTPUT	es wird eine neue Datei sequenziell erstellt, es ist nur der PUT-Makroaufruf erlaubt. Existiert bereits eine ISAM-Datei mit dem angegebenen Namen, wird sie überschrieben, der Katalogeintrag wird neu erstellt.
OUTIN	wie bei OPEN OUTPUT wird eine neue Datei erstellt, eine evtl. vorher bestehende Datei wird überschrieben. Es sind alle ISAM-Aktionen zulässig.
EXTEND	eine bestehende Datei wird sequenziell erweitert; wie bei OUTPUT sind nur Schreiboperationen mit PUT zulässig.
INOUT	eine existierende Datei soll aktualisiert werden: wie bei OUTIN sind alle ISAM-Aktionen erlaubt wie Suchen, Lesen, Ändern, Einfügen und Löschen von Sätzen.
INPUT	eine existierende Datei soll gelesen werden, es sind nur Leseoperationen zulässig.

Betriebsarten

ISAM-Dateien werden normalerweise im Move-Mode (Übertragungsbetrieb) verarbeitet; Dateiverarbeitung im Locate-Mode (Ortungsbetrieb) ist möglich, wird bei NK-ISAM allerdings nur noch aus Kompatibilitätsgründen unterstützt.

Aktionsmakro-Aufruf	**OPEN-Typ**				
	INPUT	**OUTPUT**	**EXTEND**	**INOUT**	**OUTIN**
GET	B	-	-	B	B
GETR	B	-	-	B	B
GETFL	B	-	-	B	B
GETKY	B	-	-	B	B
PUT	-	B	B	B	B
PUTX	-	-	-	B	B
INSRT	-	-	-	M	M
STORE	-	-	-	M	M
ELIM	-	-	-	x	x
SETL	x	-	-	x	x

Legende

M Move-Mode (Locate-Mode, nur wenn der Arbeitsbereich versorgt wurde)
B Move- oder Locate-Mode möglich
x Aktionsmakroaufruf zulässig
- Aktionsmakroaufruf nicht zulässig

OPEN-Fehler bei der Verarbeitung von NK-ISAM-Dateien

Zugriff aus BS2000-Version < V9.5: Der Anwender versucht eine NK-ISAM-Datei zu lesen oder zu ändern, OPEN-Fehler: DMS0DBA

Zugriff auf eine in BS2000 < V9.5 überschriebene Datei in V9.5: Wenn die Datei in einer älteren BS2000-Version überschrieben wurde, kann sie nicht mehr als NK-ISAM-Datei eröffnet werden; OPEN-Fehler DMSDDA7.

NK-ISAM-Datei auf Band: Beim Importieren/Zurückschreiben der Banddatei wurde ein falsches Datenformat im FILE- oder FCB-Makroaufruf angegeben.

NK-ISAM ist nicht geladen: Wenn im System (V9.5) NK-ISAM nicht geladen ist, wird die OPEN-Verarbeitung für NK-ISAM-Dateien mit dem Fehlercode DMS0D81 abgewiesen.

Der ISAM-Pool, in dem eine NK-ISAM-Datei eröffnet werden soll, ist überlastet: OPEN-Verarbeitung wird mit dem Fehlercode DMS0D9B abgewiesen.

Bei jedem SVC-Einstieg wird eine Überprüfung der Größe der betroffenen NK-ISAM-Datei anhand der im File Table Entry verankerten Extent-Liste durchgeführt. Wird dabei eine Dateigröße über 32 GB ermittelt und hat der Aufrufer in seinem FCB das Attribut LARGE_FILE=*FORBIDDEN gesetzt, wird die Verarbeitung abgebrochen. NK-ISAM liefert in diesem Fall den Returncode `X'00000A23'` (`FILE SIZE GROESSER 32 GIGABYTES IST NICHT ERLAUBT`).

Übernahme der Dateieigenschaften in den FCB

Die Inhalte von FCB-Feldern für Dateieigenschaften können vom Wert des entsprechenden Operanden im FILE-Aufruf abweichen; dies gilt z.B. für KEYPOS, KEYLEN, PAD und BLKSIZE. Wenn „n“ der Wert des Feldes KEYPOS und „m“ der Wert von KEYLEN im Katalogeintrag bzw. in TFT oder FCB-Makroaufruf ist, gilt für geöffnete Dateien:

– das Feld KEYLEN im TU-FCB hat den Inhalt m-1
– das Feld KEYPOS im TU-FCB hat für RECFORM = F den Inhalt (n+4)-1 = n+3

Der PAD-Wert wird nur dann berücksichtigt, wenn die Datei sequenziell mit PUT-Makroaufruf erstellt wird. Die Makroaufrufe INSRT und STORE verwenden den gesamten Speicherplatz eines Datenblocks. Bei beiden Funktionen kann es schon bei Dateierstellung zum Blocksplitting kommen.

Auch der Inhalt des Feldes BLKSIZE im TU-FCB kann vom Wert des BLKSIZE-Operanden in FILE, FCB-Makroaufruf oder Katalogeintrag abweichen: außer bei INPUT-Dateien wird der BLKSIZE-Wert unter Berücksichtigung des PAD-Wertes neu berechnet (BLKSIZE minus PAD) und im FCB-Feld BLKSIZE hinterlegt.

Beispiel

Katalogeintrag: BLKSIZE=(STD,3); PAD=15 (Standard). Der Inhalt von BLKSIZE im TU-FCB errechnet sich folgendermaßen: (3 * 2048) * (1.0 - 0.15); im Feld BLKSIZE steht während der Verarbeitung der Wert X'1467'.

DUMMY-Dateien

Für Testzwecke, vor allem für das Testen von Fehler-Routinen des Anwenderprogramms, eignen sich sog. DUMMY-Dateien, die im FILE-Makroaufruf mit *DUMMY definiert werden. Bei der Verarbeitung solcher DUMMY-Dateien finden keine Ein-/Ausgaben statt. Leseversuche führen zum Ansprung von Fehler-Routinen, Schreibaufrufe werden ignoriert, d.h. es wird in jedem Fall eine Nulloperation (d.h. keine Operation) durchgeführt.

Die folgende Tabelle zeigt, welche Ereignisse bei den ISAM-Leseoperationen auftreten.

Leseoperation	**Makro**	**Ereignis**	**EXLST-Ausgang**	**Meldung**
sequenzielles Lesen	GET/GETR	Dateiende	EOFADDR	DMS0AAE
Lesen mit Schlüssel	GETKY	Schlüssel nicht vorhanden	NOFIND	DMS0AA8
Lesen über Flags	GETFL LIMIT=KEY	Schlüssel nicht vorhanden	NOFIND	DMS0AA8
	LIMIT=END	Dateiende	EOFADDR	DMS0AAE

Blockfüllung: PAD-Wert

Bei sequenzieller Dateierstellung mit PUT kann der Anwender über den PAD-Operanden in FILE oder FCB bestimmen, wie viel Platz in den Datenblöcken freibleiben soll. Dieser Platz wird benötigt, wenn bei einer späteren Aktualisierung die Sätze in der Datei verlängert werden. Standardmäßig gilt PAD=15, d.h. 15 Prozent des Speicherplatzes im Datenblock bleiben frei. Wenn mit einem PUT-Aufruf diese Grenze erreicht wird, wird für den folgenden Satz ein neuer Datenblock angelegt.

Für NK-ISAM-Dateien gilt: sobald diese PAD-Grenze überschritten ist, wird ein neuer Datenblock angefordert; K-ISAM fordert einen neuen Datenblock bereits an, bevor die PAD-Grenze überschritten wird.

Beim sequenziellen Erstellen (PUT) nimmt der Platzbedarf einer Datei mit steigendem PAD-Faktor zu. Durch geeignete Wahl des PAD-Faktors lässt sich jedoch die nachfolgende Dateiverarbeitung (STORE/INSRT) optimieren: in den Datenblöcken ist so viel freier Platz vorzusehen, dass es bei Dateierweiterungen nicht zum Blocksplitting kommt. Zur richtigen Wahl des PAD-Faktors ist also eine Prognose erforderlich, in welchem Umfang die Datei wachsen wird.

Werden ISAM-Dateien mit STORE erstellt, hat PAD keinen Einfluss auf die Blockfüllung: STORE schreibt solange Sätze in einen Datenblock, bis dieser gefüllt ist. Wird noch ein weiterer Satz geschrieben, der nicht mehr in den Datenblock passt, kommt es zum Blocksplitting; die Blöcke werden in der Regel nur zu 50 % gefüllt.

Auch wenn die Datei zwar mit PUT, aber nicht über einen Ein-/Ausgabebereich im Programm erstellt wird, ist die PAD-Angabe wirkungslos (der Ein-/Ausgabebereich wird im FCB mit IOAREAn definiert; siehe unter „Programmpuffer = Ein-/Ausgabebereich im Benutzerprogramm“ auf Seite 77). Jeder PUT-Aufruf löst eine Schreiboperation aus, und das DVS versucht, den aktuellen Satz im letzten Datenblock unterzubringen. Erst wenn dieser gefüllt ist, legt es einen neuen Datenblock an.

Programmpuffer = Ein-/Ausgabebereich im Benutzerprogramm

Verwendet ein Programm einen eigenen Ein-/Ausgabebereich, muss dieser mindestens die Größe eines Datenblocks haben (= n * 2048 Byte, $1 \leq n \leq 16$) entsprechend BLKSIZE=(STD,n). Standardmäßig wird vom System ein Ein-/Ausgabebereich im Klasse-5-Speicher angelegt.

Die Existenz eines vom Benutzer in seinem Programm mit IOAREA1/2 definierten Ein-/Ausgabebereichs ist vor allem bei sequenzieller Verarbeitung von ISAM-Dateien von Vorteil durch die Reduzierung von SVCs:

– sequenzielles Lesen (GET/GETR): bei der ersten Leseoperation werden so viele Sätze wie möglich in den Ein-/Ausgabebereich übertragen, bevor der erste Satz dem Programm zur Verfügung gestellt wird. Bei den nachfolgenden Leseoperation werden dem Programm dann die weiteren im Ein-/Ausgabebereich vorhandenen Sätze zur Verfügung gestellt. Eine erneute Ein-/Ausgabe erfolgt erst, wenn alle Sätze gelesen sind.

– sequenzielles Schreiben (PUT): bei sequenzieller Dateierstellung oder -erweiterung werden die Sätze im Ein-/Ausgabebereich gesammelt, bis er gefüllt ist (PAD-Wert wird berücksichtigt) oder die sequenzielle Verarbeitung durch Aufruf einer anderen Operation beendet wird. Es ist also darauf zu achten, dass „PUT“-Folgen nicht durch andere Aktionsaufrufe unterbrochen werden, da dann jedes Mal aus dem Inhalt des Ein-/Ausgabebereichs ein Datenblock gebildet wird und so Datenblöcke mit geringer Blockfüllung entstehen können.

Für NK-ISAM kann im Move-Mode (Move Mode) auf den Ein-/Ausgabebereich verzichtet werden (im FCB: IOAREA1=NO). Jeder Aktionsmakroaufruf führt dann zu einem SVC.

ISAM-Zeiger

Makroaufrufe, die sich auf einen Satz beziehen, der von einem vorangegangenen Makroaufruf bearbeitet wurde, arbeiten mit internen „Zeigern“, um die aktuelle Position in der Datei bestimmen zu können:

Sowohl für den Primärschlüssel als auch für jeden in der Datei definierten Sekundärschlüssel wird ein eigener Zeiger geführt. Eine erfolgreiche Positionierungs- oder Leseoperation über einen Sekundärschlüssel (mit SETL, GET, GETKY oder GETR) verändert zunächst den Zeiger für diesen Sekundärschlüssel und setzt über den aktualisierten Sekundärschlüsselwert den Zeiger für den Primärschlüssel auf den gelesenen Satz.

Makroaufrufe, die ISAM-Zeiger auswerten, sind alle sequenziellen Makroaufrufe oder z.B. PUTX, der einen zuvor gelesenen Satz in die Datei zurückschreibt.

Kann ein Makroaufruf nicht vollständig ausgeführt werden, weil z.B. ein Fehler oder ein PGLOCK auftrat, wird der Zeiger bei NK-ISAM auf den Wert zurückgesetzt, den er vor dem Makroaufruf hatte. Eine Ausnahme bildet allerdings der Fehler „NOFIND“: da der gesuchte Satz nicht gefunden wurde, kann auf diesen Schlüssel nicht positioniert werden.

ISAM-Zeiger werden in der Regel aktualisiert bevor der Makroaufruf ausgeführt wird. Allerdings muss bei sequenziellen Leseoperationen (GET, GETR) auch der vorausgegangene Makroaufruf berücksichtigt werden: geht einer sequenziellen Leseoperation ein SETL-Makroaufruf voraus, dient dieser SETL als „Positionierungs-Makroaufruf“.

Die ISAM-Aktionen OSTAT (Open Status) und ISREQ haben keinen Einfluss auf die Zeigerposition, da sie keine Aktionen in der Datei auslösen.

Bei Dateieröffnung wird als erste Aktion für eine ISAM-Datei intern ein „SETL B“ durchgeführt, d.h. auf den ersten Satz der Datei positioniert.

Regeln für ISAM-Zeiger

Aktions-makro	Zeiger	Aktion	Bemerkung
ELIM	ohne Angabe von KEY: Zeiger wird nicht verändert; mit Angabe von KEY: Zeiger wird auf den 1. Satz dieses Schlüssels gesetzt	streicht einen Satz aus der Datei	Man kann ELIM als ' left-shift' -Operation für den Teil der Datei betrachten, der rechts des definierten Satzes steht. Daher muss bei erfolgreichem ELIM der Zeiger aktualisiert werden.
GET	setzt den Zeiger für den im Makro angesprochenen Primär- bzw. Sekundärschlüssel einen Satz weiter in Richtung Dateiende	stellt den Satz bereit, auf den der Zeiger verweist	Weist der Zeiger auf einen Satz außerhalb der Datei, wird dem Anwender die Steuerung am Ausgang EOFADDR übergeben. War der vorausgegangene Makro ein SETL oder ELIM (mit KEY), wird der Zeiger erst beim Auffinden des Satzes geändert.
GETFL	setzt den Zeiger auf den Satz, der bereitgestellt wurde, oder auf den letzten Satz des definierten Bereichs	stellt den nächsten Satz bereit, der die Markierungskriterien erfüllt	Der Satz, der auf Grund eines korrespondierenden GET oder GETR-Makroaufrufs gefunden worden wäre, ist der erste Satz, der untersucht wird.
GETKY	setzt den Zeiger auf den Satz mit dem angegebenen Primär- bzw. Sekundärschlüsselwert oder positioniert an die Stelle der Datei, wo dieser Satz stehen müsste	stellt den Satz bereit, auf den der Zeiger verweist, falls ein Satz mit dem angegebenen Schlüssel vorhanden ist	Ein GETKY für einen Schlüssel, zu dem es keinen Datensatz gibt, ist äquivalent einem GETKY für einen Satz mit der Länge null, der zwischen zwei existenten Sätzen liegen soll. Somit arbeitet ein nachfolgender GET oder GETR korrekt (siehe „GET“).
GETR	setzt den Zeiger für den im Makro angesprochenen Primär- bzw. Sekundärschlüssel einen Satz weiter in Richtung Dateianfang	stellt den Satz bereit, auf den der Zeiger verweist (-> Rückwärtslesen)	Weist der Zeiger auf einen Satz vor dem Dateianfang, erhält der Anwender die Steuerung am Ausgang EOFADDR (siehe „GET“).
INSRT	setzt den Zeiger auf die Position, die auf Grund des Satzschlüssels angegeben ist	fügt den Satz an der Stelle ein, auf die der Zeiger verweist	Der Satz wird nicht eingefügt, wenn bereits ein Satz mit diesem Schlüssel vorhanden ist. Der Zeiger verweist jedoch weiterhin auf die Position der Datei, an der der Satz eingefügt werden sollte. Ein GET nach einem nicht erfolgreichen INSRT stellt den doppelten Satz wieder bereit.

Aktions-makro	Zeiger	Aktion	Bemerkung
OPEN	setzt den Zeiger vor ersten Satz		siehe Seite 751
PUT	setzt den Zeiger genau hinter das derzeitige Ende der Datei	bringt den Satz an die Stelle auf die verwiesen wird	
PUTX	der Zeiger wird nicht verändert	bringt den Satz an die Stelle der Datei, auf die der Zeiger verweist	Es wird geprüft, ob unmittelbar vor dem PUTX ein GET-, GETR-, GETFL- oder GETKY-Makro aufgerufen wurde. So werden trotz nicht aktualisierten Zeigers Fehler vermieden.
RETRY	abhängig von der Operation, die wiederholt werden soll, oder von der Positionierungsoperation	wiederholt letzten Aktionsmakro bzw positioniert Zeiger auf Ausgangsstellung oder bringt das Programm in eine Warteschlange	RETRY selbst ändert den Zeiger nicht, es sei denn, dass es zu der Ausführung eines Aktionsmakroaufrufs führt, der seinerseits den Zeiger ändert.
SETL	*SETL B*: Zeiger wird vor den ersten Satz der Datei gesetzt *SETL E*: Zeiger wird hinter den letzten Satz der Datei gesetzt *SETL KEY*: Zeiger wird auf den ersten Satz mitdem angegebenen Primär- bzw. Sekundärschlüsselwert gesetzt oder auf den Satz mit dem nächsthöheren Schlüsselwert	keine Aktion	Die Aktualisierung des Zeigers durch ein nachfolgendes GETR oder GET wird aufgehoben. Durch das Zeigerkonzept kann bei nachfolgendem GET oder GETR ein SETL auf einen nicht vorhandenen Satz als SETL auf einen Satz der Länge null an der korrekten Position angenommen werden. Die Zeigeraktualisierung des darauf folgenden GET oder GETR wird hiernicht aufgehoben
STORE	setzt den Zeiger auf die Position, die durch den Schlüssel gegeben ist; DUPEKY=YES: beim Auftreten doppelter Schlüssel wird hinter den vorhandenen Satz positioniert	schreibt den Satz an gewünschte Stelle	Ist bereits ein Satz mit diesem Schlüssel vorhanden, wird er überschrieben, es sei denn, DUPEKY=YES wurde vereinbart. Bei DUPEKY=YES wird der neue Satz hinter den „alten“ Satz geschrieben

3.6 SAM - Sequential Access Method

SAM ist eine satzorientierte Zugriffsmethode, mit der Dateien sequenziell verarbeitet werden. Mit SAM können Sätze geschrieben, aktualisiert, und gelesen werden. SAM bietet ebenfalls eine Positionierungsfunktion, mit der auf den logischen Dateianfang, auf das logische Dateiende oder auf jeden existierenden Satz positioniert werden kann.

Als satzorientierte Zugriffsmethode übernimmt SAM für den Anwender das Blocken, Entblocken und Puffern der Sätze. Wenn im Benutzerprogramm zwei Ein-/Ausgabebereiche zur Verfügung stehen, kann Wechselpufferbetrieb genutzt werden; bei nur einem Ein-/Ausgabebereich findet keine überlappende Verarbeitung statt.

Die Zugriffsmethode SAM arbeitet überwiegend geräteunabhängig und gestattet die Verarbeitung von Dateien auf Platten und Bändern; Magnetbandkassetten werden weitgehend wie Magnetbänder behandelt.

Ab der BS2000-Version 10.0 können mit der Zugriffsmethode SAM Dateien unterschiedlicher Blockformate verarbeitet werden (siehe Kapitel „Zugriffsmethoden“, Handbuch „Einführung in das DVS“ [1]).

– K-SAM-Dateien (Key-SAM-Dateien) haben das herkömmliche Blockformat „PAMKEY“: Sie sind dadurch gekennzeichnet, dass für jede PAM-Seite DVS-Verwaltungsinformation in einem eigenen (außerhalb der Seite gelegenen) PAM-Schlüssel geführt wird.

– NK-SAM-Dateien (Nonkey-SAM-Dateien) haben das Blockformat „DATA“ oder „NO“: Sie enthalten keine gesonderten PAM-Schlüssel. Beim Blockformat „DATA“ wird die DVS-Verwaltungsinformation innerhalb der PAM-Seite in einem Blockkontrollfeld hinterlegt.
  Das Blockformat „NO“ existiert bei SAM nur für Banddateien. Im Blockformat „NO“ werden blockspezifische Verwaltungsinformationen nicht unterstützt.

Mit dem Operanden BLKCTRL in den Makros FILE und FCB kann der Anwender wählen, ob eine K- oder eine NK-Datei verarbeitet werden soll: BLKCTRL=PAMKEY vereinbart eine K-SAM-Datei, BLKCTRL=DATA oder BLKCTRL=NO legt eine NK-SAM-Datei fest.

Makroaufrufe für die Zugriffsmethode SAM

Für die Dateibearbeitung mit SAM gibt es folgende Aktionsmakroaufrufe:

Makro	Funktion
FCB	Dateisteuerblock anlegen
FEOV	für Banddateien: Bandwechsel auslösen
GET	sequenziell lesen; die Sätze werden nacheinander bereitgestellt.
PUT	sequenziell schreiben: Im Move-Mode führen die logischen Routinen der Zugriffsmethoden das Blocken der Sätze durch. Das bedeutet, dass die Ausgabe auf den Datenträger solange verzögert wird, bis der Ausgabepuffer gefüllt ist. Die Puffer werden vom System automatisch bedient. Im Locate-Mode muss der Anwender selbst für das Blocken sorgen.
PUTX	ein zuvor gelesener Satz wird zurückgeschrieben (nur im Locate-Mode bei Plattendateien)
RELSE	schließt einen Datenblock ab, d.h. für Eingabedateien: beim nächsten GET wird der nächste Datenblock eingelesen; für Ausgabedateien: beim nächsten PUT wird der Pufferinhalt als Datenblock geschrieben, der nächste Satz wird zum ersten Satz im neuen Datenblock. (Dies ist im Locate-Mode nötig, wenn der folgende Satz nicht mehr in den aktuellen Puffer passt).
SETL	in der Datei auf einen bestimmten Satz, Dateianfang oder Dateiende positionieren

OPEN-Modi

INPUT	sequenzielles Lesen in Richtung Dateiende; die Datei muss existieren
OUTPUT	neue Datei sequenziell erstellen oder eine vorhandene Datei überschreiben
EXTEND	Datei erweitern
UPDATE	nur für Plattendateien im Locate-Mode: Sätze aktualisieren; der zu aktualisierende Satz muss mit GET bereitgestellt werden, bei der Verarbeitung darf die Satzlänge nicht verändert werden, der aktualisierte Satz wird mit PUTX zurückgeschrieben
REVERSE	sequenzielles Lesen in Richtung Dateianfang; die Datei muss existieren; Banddateien, die sich über mehrere Datenträger erstecken, können nur pro Datenträger mithilfe des VSEQ-Operanden (siehe Seite 510) gelesen werden; kein automatischer Bandwechsel.

Nachfolgende Tabelle zeigt, welche OPEN-Modi bei welchen SAM-Aktionsmakroaufrufen möglich sind.

Aktionsmakroaufruf (Zugriffsmethode SAM)	**OPEN-Typ**				
	INPUT	**OUTPUT**	**EXTEND**	**UPDATE**	**REVERSE**
GET	x			x	x
PUT		x	x		
PUTX				x	
RELSE	x	x	x	x	x
SETL	x	x	x	x	x

Werden Dateien als Ausgabedateien eröffnet (OPEN OUTPUT/EXTEND), interpretiert SAM jeden PUT- oder SETL-Makroaufruf als „EOF-Anzeige“. Der letzte PUT oder SETL vor einem CLOSE zeigt dem System somit automatisch das Dateiende an. Sollen ab einem bestimmten Satz alle folgenden Sätze gelöscht werden, kann der Anwender mit dem SETL-Makroaufruf auf den gewünschten Punkt in der Datei positionieren und anschließend die Datei mit CLOSE schließen.

Für den Direktzugriff wird dem Anwender eine Wiedergewinnungsadresse bereitgestellt. Das Format dieser Wiedergewinnungsadresse ist ausführlich im Kapitel „SAM“ im Handbuch „Einführung in das DVS“ [1] beschrieben. Beim Schreiben eines Satzes wird seine Wiedergewinnungsadresse im FCB bereitgestellt. Der Anwender kann, falls er es wünscht, aus den Daten der Wiedergewinnungsadresse eine neue Datei aufbauen und hierbei eine Grundlage für anschließende nichtsequenzielle Verarbeitung der Datei schaffen, die gerade erstellt wird. Nach der Ausführung eines GET-Makroaufrufs wird diese Wiedergewinnungsadresse ebenfalls im FCB bereitgestellt. Auch hier kann der Anwender – falls er eine Datei nicht selbst erstellte – eine zweite Datei aus den Wiedergewinnungsadressen aufbauen, um eine anschließende nichtsequenzielle Verarbeitung der Datei durchzuführen.

Wird im Move-Mode mit zwei Ausgabepuffern beim Schreiben eines Datenblockes für eine Banddatei das physikalische Bandende erkannt, so wird der andere Puffer (der nur einen Satz enthält) noch auf das alte Band geschrieben. Erst dann wird der Bandwechsel eingeleitet.

Soll eine Datei im Locate-Mode erstellt werden (OPEN OUTPUT/EXTEND), erhält der Anwender nach Abschluss des OPEN in dem Register, das im FCB-Operanden IOREG angegeben ist, die Anfangsadresse des ersten zu schreibenden Satzes. Nach Ausführung des PUT-Makroaufrufs enthält das IOREG-Register die Anfangsadresse des nächsten Satzes. Bei einer Datei mit Sätzen variabler Länge (RECFORM=V) erhält der Anwender außerdem immer die Zahl der noch freien Bytes im aktuellen Block: Sie wird ihm in dem Register übergeben, das er im Operanden VARBLD des FCB-Makros angegeben hat.

Primär-/Sekundärzuweisung (Plattendateien)

Wird eine SAM-Datei erstellt oder erweitert (OPEN OUTPUT/EXTEND), müssen Primär- und Sekundärzuweisung mindestens gleich der Blockgröße sein.

Soll im Move-Mode eine Datei erstellt oder erweitert werden (OPEN OUTPUT/EXTEND), die mit RECFORM=F oder RECFORM=V definiert wurde und pro Datenblock mehr als einen Satz enthält, so gilt:

– Die Primärzuweisung muss mindestens das Doppelte der Datenblocklänge betragen (Primärzuweisung ≥ 2 * BLKSIZE). Andernfalls geht die Steuerung an den EXLST-Ausgang NOSPACE (Speicherplatzzuweisung nicht ausreichend).

– Die Sekundärzuweisung durch SAM wird bereits beim ersten Satz des letzten Blockes eingeleitet, der noch in den zugewiesenen Bereich geschrieben werden kann. Kann die Sekundärzuweisung nicht durchgeführt werden, dann geht die Steuerung an den EXLST-Ausgang NOSPACE (sofern er im Programm vorgesehen ist). Dies gibt dem Anwender die Möglichkeit, die restlichen Sätze des letzten Blockes noch zu schreiben.

Auswirkungen des LABEL-Operanden (Banddateien)

NO / NSTD	Angaben im BUFOFF-Operanden führen zum OPEN-Fehler bei CODE=ISO/OWN und BLKSIZE=STD
(STD,0)	Angaben im BUFOFF-Operanden und CODE=ISO/OWN führen zum OPEN-Fehler
(STD,1)	Angaben im BUFOFF-Operanden und die Angabe von CODE=ISO/OWN führen zum OPEN-Fehler;
(STD,2)	Standardblöcke werden in Nichtstandardblöcke (BLKCTRL=DATA/NO) und V-Sätze in D-Sätze umgewandelt; RECSIZE > 9999 führt zusammen mit RECFORM=V zum OPEN-Fehler
(STD,3)	Standardblöcke werden in Nichtstandardblöcke (BLKCTRL=DATA/NO) und V-Sätze in D-Sätze[1)] umgewandelt; RECSIZE > 9999 führt zusammen mit RECFORM=V zum OPEN-Fehler
STD / keine Angabe	Angaben im BUFOFF-Operanden führen zum OPEN-Fehler.

[1)] Bei D-Sätzen werden Längenangaben dezimal geführt.

Aus der Kombination der Angaben in den Operanden CODE, RECSIZE und RECFORM ergeben sich folgende Werte für LABEL, wenn die zu erstellende Datei die Erste auf dem Band ist:

CODE	Blockformat	RECFORM	LABEL-Wert (implizit)
EBCDIC	PAMKEY	-	(STD,1)
EBCDIC	DATA/NO	U	(STD,2)
EBCDIC	DATA/NO	F/V	(STD,3), V-Sätze -> D-Sätze
ISO/OWN	PAMKEY	-	OPEN-Fehler
ISO/OWN	DATA/NO	U	(STD,2)
ISO/OWN	DATA/NO	F	(STD,3
ISO/OWN	DATA/NO	V	V-Sätze ->D-Sätze (STD,3) OPEN-Fehler falls BLKSIZE > 9999

Die Kombination von BUFOFF und RECFORM=U oder BLKCTRL=DATA ist nicht zulässig!

Bei LABEL=(STD,1) ist es nicht möglich, eine Datei mit CODE=EBCDIC, mit Nichtstandardblöcken und D-Sätzen zu schreiben, sie kann jedoch gelesen werden.

Es kommt zum OPEN-Fehler, wenn ein im LABEL-Operanden geforderte Standardkennsatzstufe nicht mit der im VOL1-Kennsatz enthaltenen übereinstimmt.

Wurde LABEL=STD angegeben oder keine Angabe gemacht, wird die Kennsatzstufe entsprechend der o.g. Tabelle SAM-2 ermittelt. Liegt die so ermittelte Kennsatzstufe über der im VOL1-Kennsatz, gilt die im VOL1-Kennsatz definierte Kennsatzstufe. Liegt sie unter der Kennsatzstufe im VOL1-Kennsatz oder gilt aus dem VOL1-Kennsatz LABEL=(STD,0) und gilt CODE=ISO/OWN, kommt es zum OPEN-Fehler.

Satzformate für SAM-Dateien

Bei SAM sind die Satzformate F (feste Satzlänge), V (variable Satzlänge) und U (undefinierte Satzlänge) zulässig.
Bei Format U schreibt/liest SAM pro Datenblock (Puffer) nur einen Satz. Die Definition RECFORM=U in Verbindung mit BLKSIZE=STD und BLKCTRL=PAMKEY sowie einer aktuellen Satzlänge von 48 Byte hätte z.B. zur Folge, dass in jeder PAM-Seite 2000 Byte „verschwendet“ würden.

Wird ein Standardblock nicht voll ausgenutzt (z.B. nach den Aktionsmakroaufrufen RELSE, SETL, FEOV oder CLOSE), bleiben die restlichen Byte unverändert; d.h. sie haben undefinierten Inhalt.
Die Satzlänge darf die Blocklänge nicht überschreiten (siehe BLKSIZE-Operand in FILE/FCB).

Weitere Angaben zum Satzformat sind dem Kapitel „Zugriffsmethoden“, Handbuch „Einführung in das DVS“ [1] zu entnehmen.

Wiedergewinnungsadresse

Das DVS versorgt beim Erstellen einer SAM-Datei im FCB eine Wiedergewinnungsadresse, die für Positionierungen mit SETL genutzt werden kann. Die Wiedergewinnungsadresse besteht aus Block- und Satznummern. Die Blocknummer bezieht sich immer auf den logischen Datenblock (nicht auf PAM-Seiten), die Satznummer zeigt die Position des Satzes innerhalb eines Datenblocks. Bei Mehrbanddateien ist zu beachten, dass die Blocknummer nur innerhalb eines Bandes geführt wird.

Im 31-bit-TU-FCB ist die Wiedergewinnungsadresse auf zwei jeweils ein Wort lange Felder aufgeteilt: das Feld ID1BLK# enthält die Blocknummer innerhalb der Datei, das Feld ID1REC# die Satznummer innerhalb des Datenblocks. Satz- und Blockzähler werden bei PUT und GET vom System automatisch hochgezählt. Wird eine Datenübertragung ausgelöst, wird der Satzzähler automatisch zurückgesetzt.

Beim 24-bit-TU-FCB im Feld ID1RPTR in der Form „bbbbbbrr“ enthalten: „bbbbbb“ ist die Nummer des Datenblocks in der Datei, „rr“ die Nummer des Satzes innerhalb des Datenblocks. Der Satzzähler wird vom System nicht automatisch hochgezählt, dies muss der Anwender im Programm durchführen, wenn er die Wiedergewinnungsadresse nutzen will. Der Satzzähler wird jedoch bei jeder Datenübertragung automatisch zurückgesetzt.

Für Bandverarbeitung ist zu beachten, dass die Wiedergewinnungsadresse bei 24-Bit-Verarbeitung nur für Dateien mit Standardblockung versorgt wird, sodass für Dateien mit Nichtstandardblöcken kein SETL R möglich ist (siehe Makro SETL, Seite 813).

Aufbau der Wiedergewinnungsadresse:

XS-Schnittstelle: 31-bit-TU-FCB		Nicht-XS-Schnittstelle: 24-bit-TU-FCB	
ID1BLK#	ID1REC#	ID1RPTR (1 Wort)	
(1 Wort)	(1 Wort)	Byte 1-3	Byte 4
Blocknummer	Satznummer	Blocknummer	Satznummer

Der erste Satz einer Datei hat also folgende Wiedergewinnungsadressen:

im 31-bit-TU-FCB 00000001 im Feld ID1BLK# und 00000001 im Feld ID1REC#

im 24-bit-TU-FCB 00000101 im Feld ID1RPTR

Werte der Wiedergewinnungsadresse nach SETL B und SETL E, abhängig von OPEN:

OPEN TYP	**SETL B**			**SETL E**		
	ID1BLK#	**ID1REC#**	**ID1RPTR**	**ID1BLK#**	**ID1REC#**	**ID1RPTR**
INPUT, UPDATE	-	-	-	max	1	max 1
OUTPUT	1	0	1 0	Fehler	Fehler	Fehler
EXTEND	1	0	1 0	Fehler	Fehler	Fehler
REVERSE	-	-	-	-	-	-

– Feldinhalt unverändert

max höchste Blocknummer

Im Feld IDRPTR sind Block- und Satzzähler dargestellt.

Die Aktionsmakroaufrufe versorgen die Wiedergewinnungsadresse folgendermaßen:

GET

Wird durch den angegebenen Satz eine Datenübertragung erforderlich, enthält der Blockzähler die logische Blocknummer, und der Satzzähler wird zurückgesetzt. Beim 31-Bit-FCB wird der Satzzähler bei jedem Aktionsmakroaufruf aktualisiert.

PUT

Wird durch den PUT-Makroaufruf für den angegebenen Satz eine Datenüberragung ausgelöst, dann wird der Blockzähler aktualisiert und der Satzzähler zurückgesetzt. Das heißt, dass der Blockzähler auf die Nummer des neuen Datenblocks gesetzt wird, der den Satz aufnehmen soll. Bei 31-Bit-FCB wird der Satzzähler bei jedem Aktionsmakroaufruf aktualisiert.

RELSE

Wenn eine Datei erstellt oder erweitert wird (OPEN OUTPUT/EXTEND) enthält der Blockzähler die Nummer des Datenblocks, der den folgenden Satz aufnehmen soll, der Satzzähler wird auf null gesetzt.

FEOV

nach Bandwechsel werden Block- und Satzzähler für das neue Band vom System zurückgesetzt.

Beispiel

Dateieigenschaften: BLKCTRL=PAMKEY, BLKSIZE=(STD,2), RECFORM=F, RECSIZE=512

	Wiedergewinnungsadresse		
	31-bit-TU-FCB		24-bit-TU-FCB
	ID1BLK#	ID1REC#	IDRPTR
für Satz 10	00000002	00000002	00000202
für Satz 20	00000003	00000004	00000304

3.7 UPAM - User Primary Access Method

UPAM ist die primäre blockorientierte Zugriffsmethode im BS2000 für wahlfreien Zugriff auf Plattendateien. Zu jedem Zeitpunkt kann auf einen beliebigen Block der Datei lesend oder schreibend zugegriffen werden.

Auch Banddateien können mit UPAM verarbeitet werden (siehe unten).

Mit dem Operanden BLKCTRL in den Makros FILE und FCB kann der Anwender wählen, ob eine K- oder eine NK-Datei verarbeitet werden soll: BLKCTRL=PAMKEY vereinbart eine K-PAM-Datei, BLKCTRL=DATA oder BLKCTRL=NO legt eine NK-PAM-Datei fest.

Auf einer NK2-Platte wird bei Angabe der Blockgröße mit BLKSIZE=(STD,n), wobei n eine gerade Zahl ist, eine NK4-PAM-Datei angelegt; ist n eine ungerade Zahl wird eine NK2-PAM-Datei angelegt.
Auf einer NK4-Platte kann nur eine NK4-PAM-Datei liegen (siehe auch Kapitel „Zugriffsmethoden“, Handbuch „Einführung in das DVS“ [1]).

Durch die Angabe BLKSIZE=(STD,n) (n > 1) im FCB- oder FILE-Makro können mehrere 2048-Byte-Standardblöcke zu einem Datenblock (Logischen Block) zusammengefasst werden.
Bei einer K-PAM-Datei kann jeder Standardblock innerhalb des logischen Datenblockes im Programm angesprochen werden.

Bei einer NK-PAM-Datei kann nur der gesamte Datenblock (logischer Block) im Programm angesprochen werden; eine getrennte Verarbeitung der einzelnen 2048-Byte-Blöcke, aus denen er gebildet wurde, ist nicht möglich.

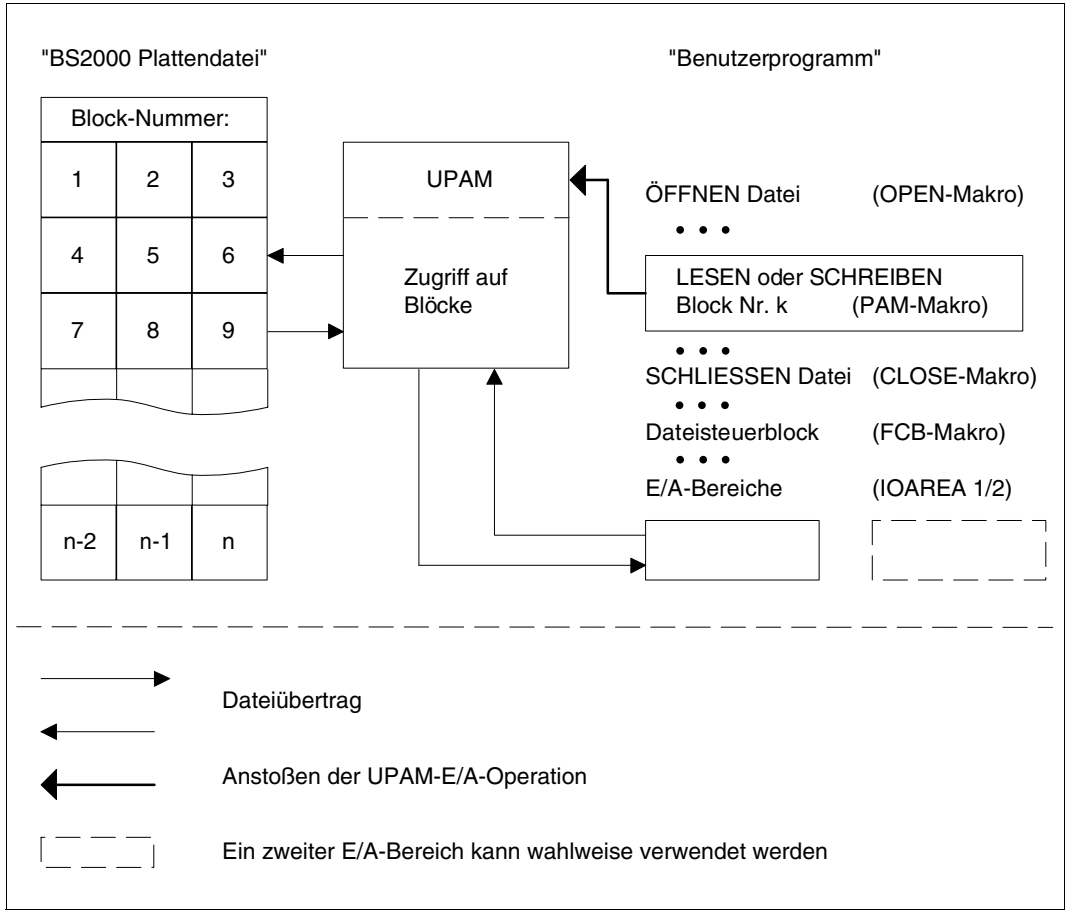


Bild 4: Arbeitsweise von UPAM

Makroaufrufe für die Zugriffsmethode UPAM

Service-Makroaufrufe

Makro	Operation	Funktion
OPEN		Datei eröffnen
CLOSE		Datei schließen
FCB		Dateisteuerblock definieren
EXLST		Fehlerausgänge definieren
PAM	CHK	Zustand der Ein-/Ausgabeverarbeitung prüfen
	LOCK	eine PAM-Seite sperren
	LRD	eine PAM-Seite sperren und den Inhalt in den Arbeitsspeicher lesen
	LRDWT	eine PAM-Seite sperren, den Inhalt in den Arbeitsspeicher lesen und Abschluss der Ein-/Ausgabe abwarten
	RD	PAM-Seite in den Arbeitsspeicher lesen
	RDEQU	wie RDWT (siehe unten); zusätzlich wird bei Dateien mit Dual Recording by Volume die Kopie aktualisiert (DRV, siehe Handbuch „DRV“ [15])
	RDWT	PAM-Seite in den Arbeitsspeicher lesen und Ein-/Ausgabe-Abschluss abwarten
	SETL	Dateizeiger positionieren
	SETLPP	Last Page Pointer (Dateiende-Zeiger) verändern; auf diese Weise können Dateien verkürzt werden, d.h. nach dem CLOSE können nicht mehr benötigte PAM-Seiten freigegeben werden. Diese Operation ist nicht zulässig für Dateien, die mit SHARUPD=WEAK/YES oder OPEN=INPUT eröffnet wurden. Bei Banddateien wird SETLPP ignoriert.
	SYNC	auf Abschluss der Ein-/Ausgabe warten und den Inhalt des Magnetbandkassettenpuffers auf Band schreiben
	UNLOCK	PAM-Seite freigeben
	WRT	aus dem Arbeitsspeicher in eine PAM-Seite schreiben
	WRTWT	aus dem Arbeitsspeicher in eine PAM-Seite schreiben und auf Abschluss der Ein-/Ausgabe warten
	WRTWU	aus dem Arbeitsspeicher in die PAM-Seite schreiben, auf den Ein-/Ausgabe-Abschluss warten und die geschriebene PAM-Seite freigeben
	WT	auf Abschluss der Ein-/Ausgabe warten

Makroaufrufe zur ereignisgesteuerten Verarbeitung

Die Makroaufrufe ENAEI, DISEI und SOLSIG müssen bei jeder Art von Ereignissteuerung aufgerufen werden; nähere Informationen zu den unten aufgelisteten Makroaufrufen sind dem Handbuch „Makroaufrufe an den Ablaufteil“ [2] zu entnehmen.

CHKEI	prüft den Zustand einer Ereigniskennung
CONTXT	greift lesend oder schreibend auf den Registersatz und Befehlszähler (den „Kontext“) eines unterbrochenen Contingency-Prozesses oder des Basisprozesses zu
DISEI	trennt einen Auftrag von einer Ereigniskennung
DISCO	entzieht einer Contingency-Definition die Möglichkeit, Contingency-Prozesse zu steuern
ENACO	ermöglicht einer Contingency-Definition die Steuerung von Contingency-Prozessen
ENAEI	weist einem Auftrag eine Ereigniskennung zu
LEVCO	ändert die Priorität eines Contingency-Prozesses oder des Basisprozesses
RETCO	beendet einen Contingency-Prozess
SOLSIG	sendet eine Anforderung an eine Ereigniskennung

OPEN-Modi

INPUT	Lesen von Blöcken aus einer vorhandenen Datei
OUTIN	Erstellen einer neuen Datei und ggf. Lesen von Blöcken aus dieser Datei
INOUT	Lesen von Blöcken aus einer vorhandenen Datei und ggf. Hinzufügen und/oder Austauschen von Blöcken

PAM-Operationen und OPEN-Modi

PAM-Makro-Funktionen	**OPEN-Modus**		
	INPUT	OUTIN	INOUT
RD, RDWT, RDEQU, LRD, LRDWT	X	X	X
WRT, WRTWT, WRTWU	-	X	X
WT, CHK, SYNC	X	X	X
LOCK, UNLOCK, SETL	X	X	X
SETLPP	-	X	X

Multi-User-Betrieb

Eine UPAM-Datei kann mit den Zugriffsmethoden UPAM, FASTPAM (siehe Seite 61) oder DIV (siehe Seite 39) erstellt bzw. bearbeitet werden. FASTPAM und DIV können jedoch nur UPAM-Dateien mit der Eigenschaft BLKCTRL=NO bearbeiten.)

Die Erlaubnis für eine parallele Dateibearbeitung ist von den bei der Eröffnung angegebenen Werten der FCB-Operanden SHARUPD, OPEN und LOCKENV abhängig. (Der FCB-Operand OPEN entspricht dem Operand MODE der Makros DIV und FPAMSRV.)

Die möglichen parallelen Eröffnungen werden in der folgenden Tabelle dargestellt:

Verträglichkeits-Matrix bei UPAM-OPEN

			USER B								
			SHARUPD =								
			*YES			*NO			*WEAK		
OPEN-Modus			INPUT	INOUT	OUTIN	INPUT	INOUT	OUTIN	INPUT	INOUT	OUTIN
USER A	SHARUPD =*YES	INPUT	X	O		X			X		
		INOUT	O	O					X		
		OUTIN	O	O					X		
	SHARUPD =*NO	INPUT	X			X			X		
		INOUT							X		
		OUTIN							X		
	SHARUPD =*WEAK	INPUT	X	X		X	X		X	X	
		INOUT							X		
		OUTIN							X		

X: OPEN erlaubt

O: OPEN nur erlaubt, wenn die Eröffner dieselbe blockorientierte Zugriffsmethode benutzen (nur UPAM/FASTPAM oder nur DIV)

und denselben Wert für den Operanden LOCKENV benutzen (alle LOCKENV=*HOST oder LOCKENV=*XCS)

und alle im selben HOST laufen *oder* in einem XCS-Verbund bei Verwendung von LOCKENV=*XCS

Anmerkungen

– Leseoperationen mit SHARUPD=*WEAK können eine Datei gleichzeitig mit jeder beliebigen Schreiboperation eröffnet haben.

  Ausnahme:
  Leseoperationen mit DIV-SHARUPD=*WEAK, die bei OPEN LOCVIEW=*NONE spezifiziert haben, besitzen dieselbe Verträglichkeit wie Leseoperationen mit UPAM-/FASTPAM-SHARUPD=*WEAK.

– Eröffner mit DIV-SHARUPD=*YES sind nicht mit Eröffnern mit UPAM-/FASTPAM-SHARUPD=*YES verträglich.

– Leseoperationen sind immer miteinander verträglich (unabhängig von Zugriffsmethode, SHARUPD-Spezifikation, LOCKENV-Spezifikation und Host).

– Nicht erlaubte Kombinationen führen zu einem OPEN-Fehler.

– Der Versuch, eine Band-Datei mit SHARUPD=YES oder WEAK zu eröffnen, führt ebenfalls zu einem OPEN-Fehler.

– Ist für eine Banddatei ohne PAM-Schlüssel BLKSIZE kein Vielfaches von 2048, so wird jeder Öffnungsversuch mit FCBTYPE=PAM ebenfalls von UPAM mit OPEN-Fehler abgewiesen.

– SHARUPD=*YES:
Bei jedem Aufruf des Allocators wird die Dateigröße überprüft. Wenn bei dieser Überprüfung eine Dateigröße ≥ 32 GB ermittelt wird und im zugehörigen FCB das Attribut LARGE_FILE=*FORBIDDEN gesetzt ist, wird die Verarbeitung abgebrochen.
UPAM liefert in diesem Fall den Returncode `X'000009AD'` (`FILE SIZE GROESSER 32 GIGABYTES IST NICHT ERLAUBT`).

UPAM-Formate

UPAM arbeitet blockorientiert, Grundlage der Verarbeitung ist bei K-PAM-Dateien der 2-KB-Standardblock, bei NK-PAM-Dateien der logische Block, dessen Größe durch den Operanden BLKSIZE im FCB- bzw. FILE-Makro festgelegt wird.
UPAM kann bis zu 16 2-KB-Standardblöcke gleichzeitig einlesen bzw. ausgegeben (LEN=(STD,n) oder LEN=n*2048 (n ≤ 16)).

Für K-PAM-Dateien gilt:

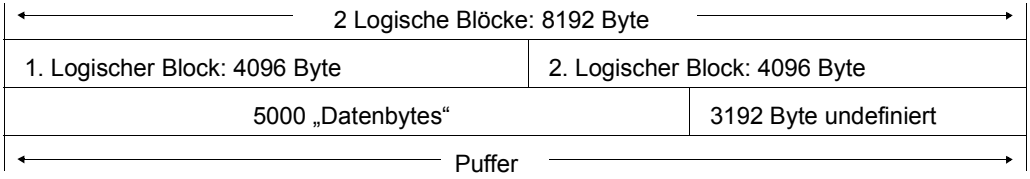
Ist der Wert des Operanden LEN im PAM-Makroaufruf kein ganzzahliges Vielfaches von 2048, wird auf das nächstgrößere ganzzahlige Vielfache von 2048 aufgerundet. Bei einer Schreiboperation ist dann der Rest der letzten zu schreibenden PAM-Seite in der Datei undefiniert. Bei einer Leseoperation wird der Rest der letzten zu lesenden PAM-Seite nicht in den Puffer übertragen.

Für NK-PAM-Dateien gilt:

Ist der Wert des Operanden LEN im PAM-Makroaufruf kein ganzzahliges Vielfaches der Größe eines Logischen Blockes, wird auf das nächstgrößere ganzzahlige Vielfache der Logischen Blockgröße aufgerundet.
Bei einer Schreiboperation ist dann der Rest des Logischen Blockes in der Datei undefiniert. Bei einer Leseoperation wird der Rest des letzten zu lesenden logischen Blockes nicht in den Puffer übertragen und der restliche Pufferinhalt ist undefiniert.

Beispiel

Für eine Datei mit BLKCTRL=NO und BLKSIZE=(STD,2) werden in einem PAM-Aufruf die Operanden WRT und LEN=5000 angegeben. Aus dem Puffer werden 5000 Byte übernommen, der Rest bis zum nächstgrößeren ganzzahligen Vielfachen der Logischen Blockgröße (8192 Byte) ist undefiniert.



UPAM für Plattendateien

Für Plattendateien bietet UPAM nachfolgende Funktionen:

Erstellen von Plattendateien - den Zugriff auf Sätze muss der Benutzer selbst programmieren (z.B. sequenziellen Zugriff oder assoziativen Zugriff mittels Hashverfahren).

Lesen von SAM- und ISAM-Dateien (OPEN INPUT) und Übertragen auf andere Datenträger (z.B. von Platte auf Band); die Eigenschaften der Dateien werden jeweils im FCB abgesetzt (z.B. BLKSIZE, RECSIZE, RECFORM). Dies ermöglicht es dem Benutzer, den Zugriff auf Sätze zu programmieren.

UPAM kann SAM- oder ISAM-Dateien nicht im UPDATE-Modus eröffnen.

ISAM-Dateien lassen sich wegen der komplexen Zusammenhänge zwischen Index- und Datenblöcken mit UPAM nicht effektiv verarbeiten. Man kann UPAM jedoch dazu benutzen, eine ISAM-Datei blockweise auf ein Band zu übertragen.

Shared-Update-Verarbeitung - mehrere parallele Aufträge können eine PAM-Datei gleichzeitig bearbeiten.
Die zulässigen OPEN-Modi bei Shared-Update-Verarbeitung einer PAM-Datei (Mono-System) sind detailliert im Kapitel „UPAM“ im Handbuch „Einführung in das DVS“ [1] beschrieben.

PAM-Makroaufrufe in Listenform (bis zu 255, die Aufrufe brauchen sich nicht alle auf die gleiche Datei zu beziehen) können mit einer UPAM-Ein-/Ausgabe-Anforderung abgearbeitet werden, d.h. es wird nur ein SVC benötigt. Die Kettung von PAM-Makros dient (ebenso wie gekettete Ein-/Ausgabe) der zeitlichen Optimierung von Benutzerprogrammen.

Benachrichtigung des Benutzerauftrags bei Beendigung einer UPAM-Ein-/Ausgabeoperation und Start eines Contingency-Prozesses (Eventing-Mechanismus)

Bei Dateien mit Dual Recording by Volume (DRV; siehe Handbuch „DRV“[15]): Information des Benutzers über den aktuellen Zustand (gegebenenfalls Kopien-Ausfall). Die Information wird bei der Ausführung einer Ein-/Ausgabe von UPAM angefordert und im FCB (Feld ID1DRVST) hinterlegt. Dieses Feld wird jedoch nur bei einer Änderung des DRV-Status aktualisiert.

Bei der Anwendung von UPAM auf eine Plattendatei ohne PAM-Schlüssel (BLKCTRL=DATA oder BLKCTRL=NO) ist Folgendes zu beachten:

– Die Datei muss Standardblöcke haben (BLKSIZE=(STD,n))
– Falls es sich nicht um eine ISAM-Datei handelt (FCBTYPE=SAM oder FCBTYPE= PAM), muss die Sekundärzuweisung mindestens so groß sein wie die vereinbarte Blockgröße (BLKSIZE).

Gekettete Ein-/Ausgabe

Gekettete Ein-/Ausgabe ermöglicht die gleichzeitige Ein-/Ausgabe von bis zu 16 logisch aufeinander folgenden PAM-Seiten in einer Datei mit einem PAM-Makroaufruf (nicht zu verwechseln mit der Verkettung von PAM-Makroaufrufen in Listenform mit dem Operanden CHAIN=). Sie vermindert so die Anzahl von Ein-/Ausgabe-Operationen (sowie Unterbrechungen) und führt zu einer Zeitersparnis bei der Verarbeitung. Auf der anderen Seite werden jedoch Arbeitsspeicherbedarf und Paging-Aufwand größer.

UPAM arbeitet mit geketteter Ein-/Ausgabe, wenn der Operand LEN im PAM-Makroaufruf einen Wert > STD bzw. > 2048 enthält.

Dateiende-Verarbeitung (EOF-Verarbeitung)

Bei einer Schreiboperation wird eine Sekundärzuweisung durchgeführt; die angegebenen PAM-Seiten werden der Datei angefügt.

Tritt bei einer Leseoperation die Dateiendebedingung auf, überträgt UPAM nur die zur Datei gehörenden PAM-Seiten in den Puffer.

UPAM informiert den aufrufenden Auftrag über die EOF-Verarbeitung folgendermaßen:

– *Ohne ereignisgesteuerte Verarbeitung:*
  Der Benutzerauftrag erhält die Steuerung am EXLST-Ausgang USERERR mit dem Fehlercode X'0922' im Feld ID1ECB des FCB. Das Feld ID1NBPP des FCB enthält die Anzahl der übertragenen PAM-Seiten. Ist der Wert dieses Feldes X'00', liegen alle zu lesenden PAM-Seiten außerhalb der Datei. Ist der Wert des Feldes größer als X'00', so hat der Benutzerauftrag eine Warte-Operation auszuführen, falls diese nicht in der Lese-Operation implizit enthalten war (d.h. in einer RDWT-Operation).

– *Mit ereignisgesteuerter Verarbeitung:*
Liegen alle zu lesenden PAM-Seiten außerhalb der Datei, übergibt UPAM dem Benutzerauftrag die Steuerung am EXLST-Ausgang USERERR mit dem Fehlercode X'0922' im Feld ID1ECB des FCB. Gehört wenigstens eine der zu lesenden PAM-Seiten zur Datei, setzt UPAM den Basisprozess fort oder startet einen Contingency-Prozess (siehe Abschnitt „TU-Eventing - ereignisgesteuerte Verarbeitung“ auf Seite 106). Jetzt enthält das Feld IDECBNPA des FECB (= File Event Control Block, siehe Seite 107) eine Anzeige, wie viele PAM-Seiten übertragen worden sind:

  X'00' alle zu lesenden PAM-Seiten wurden in den Puffer übertragen

  X'0n' n= Anzahl der PAM-Seiten, die zur Datei gehören und in den Puffer übertragen wurden

Sperren und Freigeben von PAM-Seiten

Es genügt, die Erste einer Reihe zu sperrender/freizugebender PAM-Seiten in einem PAM-Makroaufruf anzugeben; die Anzahl der zu sperrenden/freizugebenden Seiten ergibt sich aus dem Operanden LEN. Es ist jedoch zu beachten, dass der Dateizeiger nach einer LOCK- bzw. UNLOCK-Operation auf die letzte PAM-Seite verweist, die gesperrt/freigegeben wurde. Sie kann außerhalb der Datei liegen (s.o. „Dateiende-Verarbeitung“).

Eine LOCK- bzw. UNLOCK-Operation auf eine SHARUPD=NO oder SHARUPD=WEAK eröffnete Datei ist eine Nulloperation: Es wird lediglich der Zeiger auf die zuletzt bearbeitete Seite aktualisiert, dabei wird der LEN-Operand ausgewertet.

Verarbeitung von PAM-Schlüsseln

Für die Verarbeitung von PAM-Schlüsseln gibt es zwei Möglichkeiten:

– Der Benutzer liest/schreibt jeden einzelnen Schlüssel einer Reihe von PAM-Seiten: PAM-Makro-Operand MKEY=YES; Operand KEYFLD muss die Adresse eines genügend großen Bereichs angeben.

– Der Benutzer liest/schreibt nur den ersten Schlüssel einer Reihe von PAM-Seiten. Beim Schreiben wird den folgenden Blöcken derselbe Schlüssel zugeordnet wie dem ersten Block, lediglich die logische Blocknummer wird jeweils um 1 weitergezählt.

Hinweise zur UPAM-Verarbeitung von Plattendateien

Da ein Block erst bei einem expliziten Aktionsmakroaufruf in den Anwenderpuffer übertragen wird, entsteht eine Verzögerung. Deshalb muss eine asynchrone Ein-/Ausgabe mit dem Aktionsmakroaufruf WT beendet werden. Bei TU-Eventing sollte man den Makro SOLSIG (asynchron oder synchron) verwenden.

Bei jedem nicht erfolgreichen Sprung in die UPAM-Routinen wird die Steuerung an die Routine übertragen, die im EXIT-Operanden des FCB angegeben ist bzw. an die entsprechende EXLST-Routine. Im FCB wird ein Kennzeichen gespeichert.

Bei jeder von UPAM veranlassten Programmbeendigung versorgt UPAM die Register 0, 1 und 15, die in Speicherauszügen leicht auszuwerten sind.

Register 0	Adresse, an der der Abbruch auftrat
Register 1	Adresse des Elements einer UPAM-Operandenlistenkette, in dem der Fehler entdeckt wurde
Register 15	UPAM-Fehlercode

Enthält Register 1 beim ersten Aufruf des PAM-Makros eine ungültige Adresse, wird im Speicherauszug das Register 0 diese ungültige Adresse und Register 1 Nullen enthalten (d.h. der Fehler trat auf, bevor das erste Element der Operandenlistenkette gefunden wurde).

UPAM verwendet folgende EXLST-Ausgänge:

ERRADDR	Hardwarefehler oder abnormale Ein-/Ausgabebeendigung
USERERR	Unzulässige Makroanwendung im Programm oder Lesezugriff auf eine nicht zur Datei gehörende PAM-Seite (Dateiende)
EOFADDR	Versuch, eine *DUMMY-Datei zu lesen
PGLOCK	Nicht alle angeforderten Sperren sind innerhalb der gegebenen Zeit verfügbar, und der Auftrag hält momentan keine Sperren
DLOCK	Die Anforderung einer Sperre wird abgewiesen und der Auftrag hält bereits Sperren

PAM-Seiten, die einer Datei zugewiesen, aber vom Eigentümer dieser Datei noch nicht geschrieben wurden, sind an ihrem systemintern verschlüsselten Dateinamen (CFID=Coded File ID, Byte 0-3 des PAM-Schlüssels bzw. des Blockkontrollfeldes) zu erkennen, der dann mit dem aktuellen Dateinamen nicht übereinstimmt. Der Vergleich ist vom Benutzer durchzuführen. Dabei ist Folgendes zu beachten (siehe auch Operand KEYFLD im PAM-Makroaufruf, Seite 759):

– Zum OPEN-Zeitpunkt wird die aktuelle CFID in das erste Wort im Feld ID1KEY1 des FCB geschrieben.

Die folgenden Punkte gelten nur bei der Verarbeitung von K-PAM-Dateien (BLK-CONTR=PAMKEY):

– Nach Ausführung einer RDWT-, LRDWT- oder RDEQU-Operation steht die CFID des gelesenen Blocks im ersten Wort des FCB-Feldes ID1KEY2.

– Nach Ausführung einer der Operationen WRT, WRTWT, WRTWU, WT steht die CFID der betreffenden PAM-Seite im ersten Wort des FCB-Feldes ID1KEY1. Der von OPEN erstellte Eintrag wird überschrieben; er sollte daher vor der Verarbeitung für spätere Vergleiche sichergestellt werden.

– Nach Ausführung der Operationen LRD und RD ist der Inhalt der Felder ID1KEY1 und ID1KEY2 unverändert.

– Bei ereignisgesteuerter Verarbeitung steht nach dem Abschluss einer Ein-/Ausgabeoperation die CFID der betreffenden PAM-Seite im ersten Wort des FCB-Feldes ID1KEY1.

Die Felder ID1LWB (PARMOD=24) bzw. ID1LWBPT (PARMOD=31) im FCB enthalten die Adresse des letzten Blocks, auf dem von UPAM eine Ein-/Ausgabeoperation erfolgreich durchgeführt wurde. Als Anzeige für einen evtl. noch ausstehenden WT wird das linke Byte des Feldes ID1LWB verwendet.

War die letzte UPAM-Operation für die Datei ein erfolgreicher WT, erhält die Anzeige in ID1LWB den Wert X'00' und die drei niederwertigen Byte des Feldes ID1LWB bzw. das Feld ID1LWBPT enthalten die Adresse des Blocks, auf den der WT sich bezog. Dabei spielt es keine Rolle, ob die den WT auslösende Operation erfolgreich abgeschlossen wurde.

Hat die letzte UPAM-Operation für die Datei keinen WT ausgelöst, wird die Anzeige in ID1LWB auf den Wert X'FF' gesetzt. Der Inhalt der rechten drei Byte von ID1LWB bzw. ID1LWBPT ist dann ohne Bedeutung.

UPAM-Verarbeitung von Banddateien

Für Banddateien bietet UPAM folgende Funktionen:

Erstellen von Banddateien, die nicht über ein Band hinausgehen. Den Zugriff auf logische Sätze dieser Dateien muss der Benutzer selbst programmieren.

Lesen von SAM-Dateien mit Standardblöcken - die Dateieigenschaften werden durch die OPEN-Verarbeitung (siehe Kapitel „OPEN-Verarbeitung“, Handbuch „Einführung in das DVS“ [1]) abgesetzt, z.B. BLKSIZE, RECSIZE, RECFORM. Dies ermöglicht es dem Benutzer, Satzzugriff zu programmieren.

Gekettete Ein-/Ausgabe ist bei Banddateien nicht möglich!

Benachrichtigung des Benutzerauftrags bei Beendigung einer UPAM-Ein-/Ausgabeoperation und Start eines Contingency-Prozesses (Eventing-Mechanismus).

Grundsätzlich ist bei der Anwendung von UPAM zu beachten:

– UPAM ist eine blockorientierte Zugriffsmethode, d. h. eine logische Struktur innerhalb der Blöcke ist dem System nicht bekannt. Die Satzverarbeitung muss vom Anwender selbst programmiert werden.

Bei der Anwendung von UPAM auf eine Banddatei ist Folgendes zu beachten:

– Die Datei muss auf einem einzigen Band Platz finden.

– Jede Schreib-/Leseoperation bearbeitet genau einen physikalischen Block.

– Bei einer Datei vom Format BLKCTRL=PAMKEY (explizit oder implizit) muss dies ein Standardblock der Länge 2064 Byte sein: In den ersten 16 Byte steht der PAM-Schlüssel, die restlichen 2048 Byte enthalten die Anwenderdaten. Der Operand LEN des PAM-Makros darf nur die Werte STD oder 2048 annehmen.

– Bei einer Datei vom Format BLKCTRL=DATA oder BLKCTRL=NO kann man BLKSIZE in Byte angeben, wobei der Wert ein Vielfaches von 2048 sein muss. Der Wert für LEN darf den für BLKSIZE nicht übersteigen. Auf das Band werden Blöcke der Größe LEN geschrieben, die entweder nur Anwenderdaten (bei BLKCTRL=NO) oder 12 Byte Blockkontrollfeld und LEN - 12 Byte Anwenderdaten (bei BLKCTRL=DATA) enthalten. Diese Blöcke können mit der UPAM-Funktion RD gelesen werden, wobei für den Wert LEN_{RD} von LEN im Leseaufruf gilt:
  $LEN_{WR} \leq LEN_{RD} \leq$ BLKSIZE. (LEN_{WR}: Wert von LEN beim Schreiben der Datei).

  Bei einer Datei mit FCBTYPE=ISAM ist in diesem Zusammenhang folgende Besonderheit zu beachten: Unabhängig von der BLKSIZE-Angabe arbeitet UPAM stets mit der Blockgröße 2048 Byte, da bei ISAM jeder 2K-Block ein Blockkontrollfeld besitzt und daher eine eigene Einheit darstellt. In diesem Fall darf der Wert für LEN 2048 nicht übersteigen.

Eine Banddatei ohne PAM-Schlüssel, die mit UPAM in V10.0 erstellt wurde, kann auf Grund des andersartigen Blockformats in BS2000-Versionen < V10.0 nicht mit UPAM gelesen oder bearbeitet werden. Eine Bearbeitung mit BTAM ist jedoch möglich. Dies gilt auch für NK-ISAM-Dateien, die in V10.0 erstellt und in V9.5 eingespielt werden sollen.

– Bei UPAM-Zugriff auf eine Banddatei mit BLKCTRL=PAMKEY und Blöcken, die vom Standardformat (2064 Byte) abweichen, können folgende Fehler auftreten:
  – Hardware-Fehler (Fehlercode = 927);
  – die Daten werden im Puffer falsch abgespeichert;
  – der PAM-Schlüssel wird verfälscht.
– Auf Banddateien darf nicht von mehreren Aufträgen, bzw. von einem Auftrag mehrfach zugegriffen werden (was auf die Eigenschaften der Magnetbänder zurückzuführen ist); das heißt:
  – eine Banddatei kann nicht mit SHARUPD=YES oder WEAK eröffnet werden;
  – eine Banddatei, die bereits eröffnet ist, kann nicht nochmals eröffnet werden, auch wenn beide Eröffnungen vom Typ INPUT sind!
  – eine Banddatei kann nicht mit einem FCB-Operanden PAMREQS > 1 eröffnet werden.
– Es ist möglich, mit UPAM eine Banddatei im wahlfreien Zugriff zu lesen. Der Zeitaufwand hierfür kann aber beträchtlich sein.
– Es ist möglich, ab einer bestimmten Stelle einer bestehenden Banddatei PAM-Blöcke zu schreiben. Der letzte neu geschriebene Block wird automatisch zum letzten Block der Datei, auch wenn die bestehende Datei mehr Blöcke enthielt. Die Datei kann dann nur noch bis zu diesem Block gelesen werden. Wird anschließend ein (weiter vorne liegender) Block der Datei gelesen und die Datei daraufhin geschlossen, so wird der zuletzt gelesene Block zum letzten Block der Banddatei.

Hinweise zur Programmierung

Magnetbandkassetten werden wie Magnetbänder behandelt.

Bei jeder von UPAM veranlassten Programmbeendigung bringt UPAM die Adresse, die den Abbruch verursacht hat, in das Register 0, die Adresse des Elements der UPAM-Operandenlistenkette, in dem der Fehler entdeckt wurde, in das Register 1 sowie den UPAM-Fehlercode in das Register 15. So kann diese Information in Speicherauszügen leicht gefunden werden.

Enthält Register 1 nach dem ersten PAM-Makroaufruf eine ungültige Adresse, ist diese Adresse im Speicherauszug im Register 0 zu finden; das Register 1 hat dann den Wert 0, d.h. der Fehler trat auf, bevor das erste Element der Operandenlistenkette gefunden wurde.

UPAM verwendet folgende EXLST-Ausgänge:

ERRADDR Hardwarefehler oder abnormale Ein-/Ausgabe-Beendigung

USERERR unzulässige Operation, z.B. Lesezugriff auf eine nicht zur Datei gehörende PAM-Seite (Dateiende)

Die Felder ID1LWB (PARMOD=24) bzw. ID1LWBPT (PARMOD=31) im FCB enthalten die Adresse des letzten Blocks, auf dem von UPAM eine WT-Operation erfolgreich durchgeführt wurde. Als Anzeige wird das linke Byte des Feldes ID1LWB verwendet.

War die letzte UPAM-Operation für die Datei ein erfolgreicher WT, erhält die Anzeige in ID1LWB den Wert X'00' und die drei niederwertigen Byte des Feldes ID1LWB bzw. das Feld ID1LWBPT enthalten die Adresse des Blocks, auf den der WT sich bezog. Dabei spielt es keine Rolle, ob die den WT auslösende Operation erfolgreich abgeschlossen wurde.

Hat die letzte UPAM-Operation für die Datei keinen WT ausgelöst, wird die Anzeige in ID1LWB auf den Wert X'FF' gesetzt. Der Inhalt der rechten drei Byte von ID1LWB bzw. ID1LWBPT ist dann ohne Bedeutung.

Kettung von PAM-Makroaufrufen in Listenform

PAM-Makroaufrufe, die miteinander verkettet werden sollen, sich aber nicht notwendig auf die gleiche Datei beziehen müssen, sind mit dem Operanden MF=L in Listenform zu erzeugen und in einem Konstantenbereich unterzubringen; die Kettung wird durch Angabe des Operanden CHAIN= erreicht.

Die Makroaufrufe haben (bis auf den Letzten) folgendes Format:

$element_n$	PAM	fcbadr,operation,...,MF=L,CHAIN=$element_{n+1}$

Beim letzten Element der Kette entfällt der Operand CHAIN.

Der Aufruf einer Kette von PAM-Makroaufrufen in Listenform erfolgt mit einem PAM-Makro der folgenden Form:

	PAM	MF=(E,$element_1$)

Es wird jeweils nur ein SVC pro Kette ausgeführt, d.h. man vermeidet durch Kettung von UPAM-Anforderungen den Aufwand einer mehrfachen SVC-Bearbeitung.

Beispiel zur Codierung

```
        START
        LDBASE 10
        USING *,10
        .
        .
        PAM MF=(E,ELEM1)
        .
        .
        .
        TERM

*KONSTANTENBEREICH
ELEM1   PAM     ......,MF=L,CHAIN=ELEM2
ELEM2   PAM     ......,MF=L,CHAIN=ELEM3
ELEM3   PAM     ......,MF=L
        .
        .
        .
        END
```

Bei fehlerfreiem Ablauf erhält der Anwender die Kontrolle bei der Anweisung, die auf den PAM-Makroaufruf folgt, der die Abarbeitung einer Operandenlisten-Kette forderte.

Alle Operationen werden in genau der Reihenfolge ausgeführt, in der die PAM-Makro-Listen innerhalb der Kette vorkommen - mit einer Ausnahme: Wenn die erste Operation, die eine Sperre fordert, erkannt wird, wird der Rest der Kette durchgeprüft, und alle Operationen, die Sperren fordern, werden registriert. Falls innerhalb der angegebenen Zeit nicht alle geforderten Sperren verfügbar sind, wird die Kette bei der Operation abgebrochen, die die erste Sperre forderte.

Wird eine in der Kette angeforderte Aktion nicht erfolgreich durchgeführt, werden auch die auf diese Anforderung folgenden Aktionen nicht ausgeführt (einschließlich Sperren). Die Steuerung wird an den entsprechenden EXLST-Ausgang übergeben; das Register 1 weist auf den FCB der fehlerauslösenden Datei. Der Fehler-Code (ID1ECB) und das Fehlerbyte (ID1XITB) werden in diesem FCB wie üblich gesetzt. Das ID1CHERR-Feld im FCB wird auf die Adresse desjenigen Elements der Operandenlisten-Kette gesetzt, in dem der Fehler auftrat.

Eine Prüfoperation auf eine noch nicht abgeschlossene Ein-/Ausgabe-Operation führt ebenfalls dazu, dass die Kontrolle aus der Operandenlisten-Kette an das Benutzerprogramm an der angegebenen Adresse übergeben wird. Die restlichen Anforderungen in der Kette (einschließlich Sperrungen) werden nicht ausgeführt, und das ID1CHERR-Feld im FCB wird auf die Adresse desjenigen Elements der Operandenliste gesetzt, das den CHK enthält. Es ist also davon abzuraten, Prüfvorgänge in Operandenlisten-Ketten abzusetzen.

Der Anwender muss sicherstellen, dass innerhalb einer Kette die Operationen Sperren, Lesen, Schreiben, Warten, Prüfen und Freigeben sinnvoll angewendet werden. Puffer und Schlüsselfelder werden von UPAM entsprechend der Anforderung benutzt. Die Existenz eines Puffers und die Zugriffsbefugnis werden überprüft, aber es gibt keine Garantie, dass ein Puffer oder ein Schlüsselfeld, die von einer Operation in einer Kette gefüllt werden, von einer späteren Operation dieser Kette nicht überschrieben werden.

Ein Pseudo-Programmabschnitt (DSECT), der mit dem Makroaufruf IDPPL generiert werden kann, beschreibt das Format der PAM-Operandenliste.

In allen Fällen, in denen eine Kette von UPAM-Operandenlisten nicht vollständig abgearbeitet werden kann (z.B. eine Ein-/Ausgabe-Operation ist gescheitert, ein Fehler wurde entdeckt, eine Sperre konnte nicht ausgeführt wurden, die EOF-Bedingung trat auf oder ein CHK wurde an eine laufende Ein-/Ausgabe-Anforderung gestellt), wird in das ID1CHERR-Feld des FCB die Adresse des ersten nicht ausgeführten Eintrags der Kette gebracht. Der Anwender kann davon ausgehen, dass alle Einträge davor richtig ausgeführt wurden.

UPAM meldet über FCB-EXIT und EXLST keinen Fehler,

– wenn der UPAM-SVC ausgeführt wird und weder Register 1 noch ein Operand der Kette eine gültige Adresse enthalten (z.B.: die Adresse liegt nicht an einer Wortgrenze oder beschreibt ein Feld, das nicht ganz zum virtuellen Adressraum des Anwenders gehört und nicht groß genug ist, um eine UPAM-Makro-Operandenliste aufzunehmen).

– wenn die FCB-Adresse in der UPAM-Makro-Operandenliste fehlt oder ungültig ist.

In beiden Fällen gibt es keinen FCB, dem der Fehler gemeldet werden könnte, deshalb veranlasst UPAM eine vorzeitige Beendigung des Programms.

Eine UPAM-Operandenlisten-Kette wird vollständig für gültig erklärt, bevor eine Aktion ausgelöst wird. Ist eine CHAIN-Adresse oder eine FCB-Adresse ungültig, wird der Auftrag vorzeitig beendet, bevor irgendein Element der Kette ausgeführt wird.

Wird der Eventing-Mechanismus verwendet und steht bei Ende der Ein-/Ausgabe keine Ereigniskennung zur Verfügung, an die das Ereignis gemeldet werden kann, wird das Benutzerprogramm ebenfalls abgebrochen.

TU-Eventing - ereignisgesteuerte Verarbeitung

Die im Folgenden beschriebene ereignisgesteuerte Verarbeitung sowie die genannten Makroaufrufe sind ausführlich im Handbuch „Makroaufrufe an den Ablaufteil“ [2] dargestellt.

Ereignisgesteuerte Verarbeitung wird von UPAM dazu benutzt, einem Auftrag die Beendigung einer angeforderten Ein-/Ausgabe mitzuteilen. Der Auftrag kann

– parallel zur UPAM-Ein-/Ausgabe fortgesetzt werden und bei Eintreten des erwarteten Ereignisses (hier die Beendigung der angeforderten Ein-/Ausgabe) mit einem Contingency-Prozess fortfahren (asynchrone Verarbeitung).

– auf die Beendigung der angeforderten Ein-/Ausgabe warten und dann fortfahren (synchrone Verarbeitung, die natürlich auch ohne ereignisgesteuerte Verarbeitung möglich ist).

UPAM gibt nach Vollendung einer Ein-/Ausgabe-Operation eine Meldung an die zugeordnete Ereigniskennung (mit dem Makroaufruf POSSIG). Diese Meldung trifft dann sofort oder später auf die vom Benutzer ausgegebene Anforderung (SOLSIG-Makroaufruf). Liegen sowohl Anforderung als auch Meldung vor (für dieselbe Ereigniskennung), wird ein Contingency-Prozess gestartet oder der Basisprozess fortgesetzt.

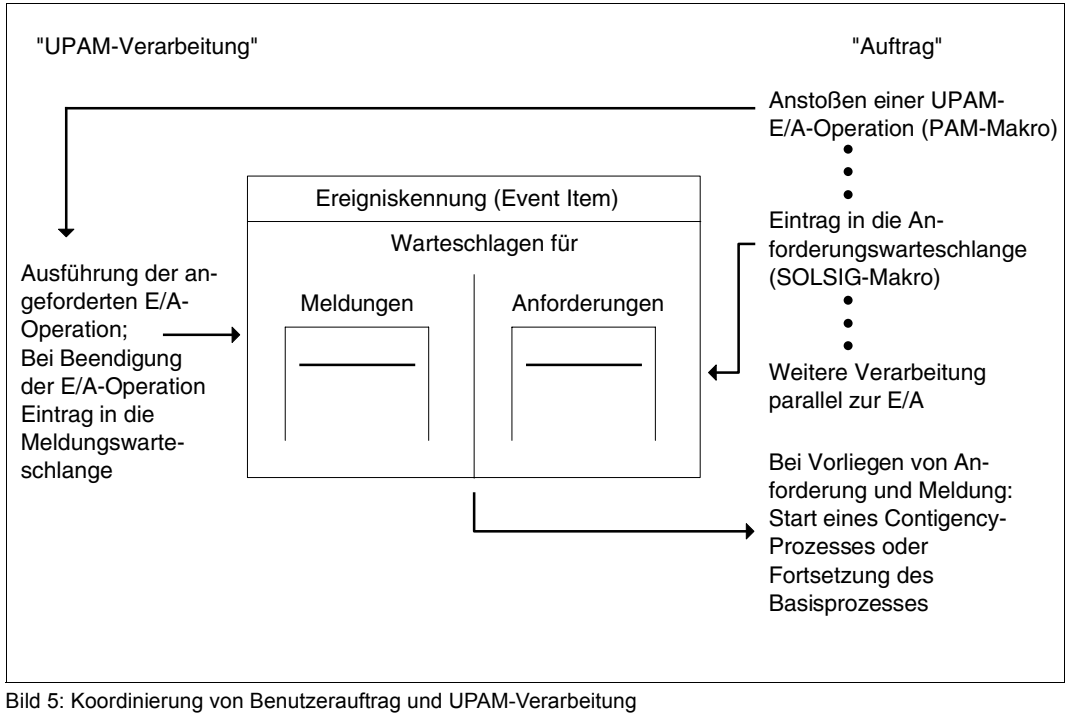


Bild 5: Koordinierung von Benutzerauftrag und UPAM-Verarbeitung

Basisprozess

Dem System müssen die zu verwendenden Ereigniskennungen und Contingency-Definitionen mitgeteilt werden (Makroaufrufe ENAEI, ENACO).

Für jede Ein-/Ausgabe muss dem System die Adresse eines FECBs (File Event Control Block) mitgegeben werden. Parallel laufende Ein-/Ausgaben müssen auf verschiedene FECBs verweisen. Die maximale Anzahl paralleler Ein-/Ausgaben wird durch den FCB-Operanden PAMREQS festgelegt; für Banddateien gilt: PAMREQS=1.

Die Zahl der Contingency-Definitionen richtet sich danach, ob unterschiedliches Vorgehen nach einer ausgeführten Ein-/Ausgabe-Operation gewünscht ist. Soll z.B. immer auf die gleiche Weise verfahren werden, genügt es, die Contingency-Definition nur einmal zu codieren.
Für jede Ereigniskennung muss ein 14 Byte langer Steuerblock eingerichtet werden - der FECB = File Event Control Block.
Für jede Datei muss im FCB-Makroaufruf der Operand PAMREQS= korrekt angegeben werden. PAMREQS legt die Höchstzahl gleichzeitig für diese Datei anzufordernder Ein-/Ausgabe-Operationen fest.
Solange die erste Ein-/Ausgabe-Operation mit einem FECB nicht beendet ist, darf dieser FECB nicht für andere Ein-/Ausgabe-Operationen benutzt werden.

Bei jeder UPAM-Ein-/Ausgabe-Anforderung muss die Adresse des zugeordneten Steuerblocks angegeben werden (Operand FECB= im PAM-Makroaufruf). Es dürfen durch den PAM-Makroaufruf keine Warteoperationen angefordert werden, weder explizit noch implizit. Die Befehlsfolge Lesen (RD) → Schreiben (WRT) → Warten (WT) auf denselben Block führt demnach zu einem undefinierten Ergebnis. Das Ereignis (Beendigung der Ein-/Ausgabe) muss vor dem WRITE-Aufruf abgewartet werden.

Nach jeder UPAM-Ein-/Ausgabe-Anforderung muss genau eine Anforderung an die zugeordnete Ereigniskennung abgegeben werden (Makroaufruf SOLSIG). Mit dieser Anforderung kann festgelegt werden, ob der Basisprozess parallel zur Ein-/Ausgabe weiterläuft oder deren Beendigung abwartet.

Beim Start eines Contingency-Prozesses werden diesem in den Registern 2 und 3 folgende Informationen übergeben:

bei PARMOD=31: die Ein-/Ausgabe wird über die 31-Bit-Operandenliste angestoßen, die Informationen werden in den Registern 2, 3 und 4 übergeben:

Register	Information
2	enthält den Ereignis-Informationscode
3	enthält in den beiden rechten Byte einen vom Anwender beim Anstarten der Ein-/Ausgabe mitgelieferten POST-CODE und im linken Byte ein Kennzeichen für UPAM-Event (X'10')
4	enthält die Adresse der Operandenliste der beendeten Operation

bei PARMOD=24: Die Ein-/Ausgabe wird über die „alte“ 24-Bit-Operandenliste angestoßen, die Informationen befinden sich in den Registern 2 und 3:

Register	Information
2	enthält den Ereignis-Informationscode
3	enthält in den drei rechten Byte die Adresse der Operandenliste der beendeten Operation und im linken Byte den Wert X'10'

Wird über eine mit PARMOD=31 erzeugte PAM-Operandenliste ein mit PARMOD=24 erzeugter SOLSIG-Makro oder eine 24-Bit-Contingency-Definition angesprochen, enthält der sekundäre Returncode (linkes Byte von Register 15) einen Hinweis auf inkonsistente Längen von Sender und Empfänger.

Aufbau des Steuerblocks FECB (File Event Control Block)

Der FECB muss auf Wortgrenze ausgerichtet werden. Er kann mit dem Makroaufruf IDECB mit symbolischen Namen versehen werden.

Ablaufteil-Markierungsbyte

Feldbedeutung	Feldlänge (Byte)	Feldname
interne Kurzkennung der Ereigniskennung	4	CBEVID
Adresse des FCB	4	CBP1LNK
Standard-Gerätebyte	1	CBSDB
Fehlerbytes	3 x 1	CBSB1, CBSB2, CBSB3
Ablaufteil-Markierungsbyte	1	CBEFB
Anzahl übertragener PAM-Seiten	1	CBNPA

Eine UPAM-Ein-/Ausgabe-Operation kann auf verschiedene Arten beendet werden (AMB: Ablaufteil-Markierungsbyte, siehe FECB):

normale Ein-/Ausgabe-Beendigung	AMB=X'80'
Ein-/Ausgabe-Operation führte zu Sonderbedingung	AMB=X'C0'
nicht behebbarer Fehler (z.B. Hardwarefehler)	AMB=X'A0'

In einem Contingency-Prozess kann der Anwender auf die einzelnen Möglichkeiten der Ein-/Ausgabe-Beendigung entsprechend reagieren.

3.8 Dateien größer 32 GB

Ab BS2000/OSD-BC V5.0 unterstützt BS2000 Dateien und Volumes mit einer Kapazität bis zu 4 Terabyte. Diese Dateien und Volumes werden „große Dateien“ und „große Volumes“ genannt.

Die von BS2000 unterstützten Grenzwerte betragen jetzt:

– die maximale Kapazität einer einzelnen Platte ca. 4 TB (2.147.483.647 PAM-Seiten)
– die maximale Dateigröße ebenfalls ca. 4 TB (2.147.483.647 PAM-Seiten)

Große Dateien und große Volumes werden nur in speziellen Pubsets unterstützt, die für die Verwendung dieser großen Objekte vom Systembetreuer attributiert werden müssen. Solche Pubsets können in BS2000/OSD-BC -Versionen < V5.0 nicht importiert werden.

Erweiterung des Katalogeintrags

Zentral für die Aufhebung der 32-GB-Grenze für die Volume- und Dateigröße ist die Einführung von 4-Byte-Feldern für folgende im Katalogeintrag abgelegte Daten:

– FILE-SIZE, der für die Datei allokierte Speicherplatz
– HIGHEST-USED-PAGE, der davon aktuell durch Daten belegte Speicherplatz
– LHP (Logical halfpage number) und PHP (physical halfpage number) der einzelnen Extents in der Extent-Liste, die den logischen Halbseiten „physikalische“ Halbseiten von Volumes zuordnet.

3-Byte- und 4-Byte-Felder

Blocknummern und Blockzähler sind an verschiedenen Benutzerschnittstellen des BS2000 sichtbar. Während alle neueren Ausprägungen dieser Schnittstellen konsequent 4-Byte-Felder verwenden, werden bei manchen älteren Schnittstellenversionen 3-Byte-Felder verwendet. Hier ist beim Vorhandensein von Dateien ≥ 32 GB und in seltenen Fällen auch bei Volumes ≥ 32 GB mit Kompatibilitätsproblemen zu rechnen.

Neues Format für die Extent-Liste

Die Einführung von 4-Byte-LHPs und 4-Byte-PHPs bedeutet, dass für die Extent-Liste ein neues (zusätzliches) Format eingeführt wird.

Der Zusammenhang zwischen der maximalen Größe von Datenträgern und Dateien und der Feldbreite von LHP und PHP wird in Bild 6 dargestellt:

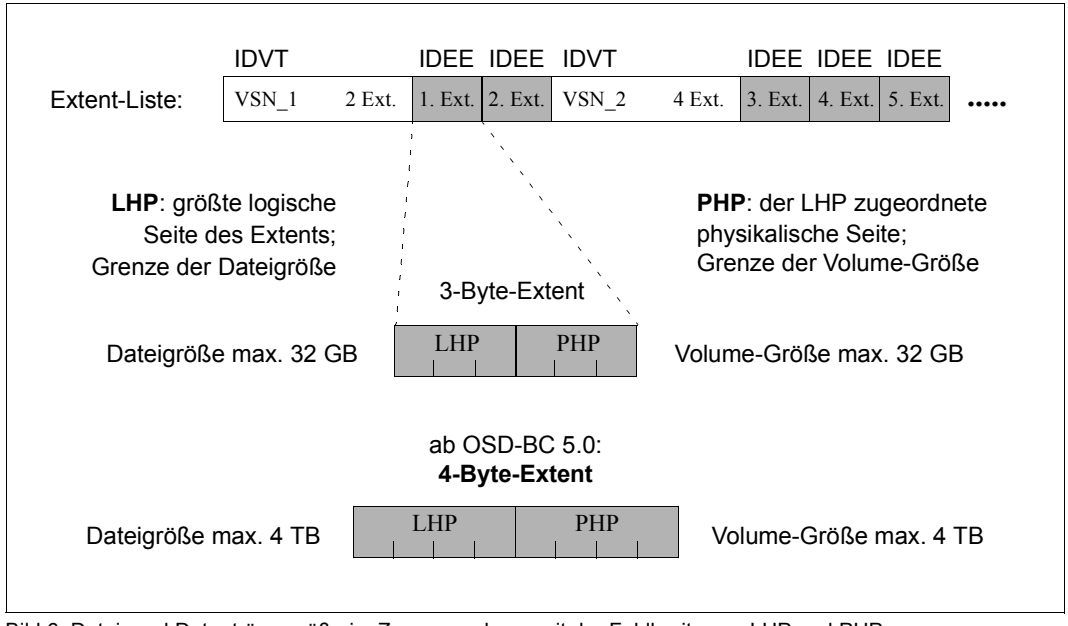


Bild 6: Datei- und Datenträgergröße im Zusammenhang mit der Feldbreite von LHP und PHP

Dieses neue Format ist in älteren BS2000/OSD-BC-Versionen nicht interpretierbar. Um so weit wie möglich Abwärtskompatibilität sicherzustellen, unterstützt BS2000 beide Formate der Extent-Liste:

– Grundsätzlich wird das „alte“ Format mit 3-Byte-Blocknummern verwendet.

– Nur im Fall von großen Dateien oder von Dateien, für deren Adressierung PHPs > X'FFFFFF' benötigt werden, wird das neue Format mit 4-Byte-Blocknummern verwendet.

Extent-Listen enthalten also entweder Extents mit 3-Byte-Blocknummern oder Extents mit 4-Byte-Blocknummern.

Einschränkungen für große Dateien

- Auf dem Home-Pubset dürfen keine großen Dateien liegen.
- Die Dump-Datei $TSOS.SLEDFILE (SLED-Datei) darf keine Datei ≥ 32 GB sein.
- Die Paging-Datei darf keine Datei ≥ 32 GB sein.
- Eine SYSEAM-Datei darf keine Datei ≥ 32 GB sein.
- SIR unterstützt beim Einlesen von ARCHIVE-Bändern keine großen Dateien.
- Dateien mit BLKCTRL=PAMKEY
  Im Systemteil des Pamkey ist die logische Seitennummer als 3-Byte-Feld hinterlegt. Dies kann nicht für alle Zugriffsmethoden geändert werden.

Übersicht über die 32-GB-relevanten DVS-Makro-Schnittstellen

Schnittstelle	**Änderung**
FCB	neuer Operand für große Dateien
FILE	neuer Operand für große Dateien
FSTAT	Prüfungs- und Umstellungsaufwand bei VERSION=0/1
OPEN	Semantikproblem beachten
RDTFT	Ausgabe des Dateiattributes „große Datei“
DIV	neuer Operand für große Dateien; vergrößerter Wertebereich für BLOCK und SPAN
FPAMACC	vergrößerter Wertebereich für BLOCK
FPAMSRV	neuer Operand für große Dateien

Benutzerprogramme

Wie bereits erwähnt, kann man nicht davon ausgehen, dass alle Programme für den Zugriff auf große Objekte vorbereitet sind, d.h. mit 4 Byte breiten Blocknummern und Blockzählern zurecht kommen, wobei Benutzerprogrammen nur Schnittstellen für Zugriff auf und Bearbeitung von Dateien und deren Metadaten zur Verfügung stehen.

Dabei lässt sich das Verhalten von Programmen wie folgt klassifizieren:

Klasse A: Ein Programm ist in der Lage, große Dateien ohne Einschränkung zu verarbeiten. Dieses Verhalten wird als LARGE_FILES-fähig bezeichnet.

Klasse B: Ein Programm ist zwar nicht auf die Bearbeitung großer Dateien und/oder ihrer Metadaten vorbereitet, ist aber in der Lage, entsprechende Zugriffe, die als fehlerhaft betrachtet werden müssen, definiert abzuweisen oder es erfolgen im Programm keine Zugriffe auf Dateien und ihre Metadaten. Dieses Verhalten wird als LARGE_FILES-kompatibel bezeichnet.

Klasse C: Ein Programm ist nicht auf die Bearbeitung großer Dateien vorbereitet und ist auch nicht in der Lage, entsprechende Zugriffe definiert abzuweisen. Dieses Verhalten wird als LARGE_FILES-inkompatibel bezeichnet.

Für Konfigurationen, die große Dateien beinhalten, müssen LARGE_FILES-kompatible oder -fähige Programme vorausgesetzt werden. Dabei ist als Regelfall LARGE_FILES-Kompatibität zu sehen. Das Wachstum über 32 GB hinaus dürfte sich vorerst auf relativ wenige Dateien beschränken; nur Programme, die auf diese zugreifen, müssen LARGE_FILES-fähig sein.

Eine Konkretisierung dieser Klassifizierung für die relevanten DVS-Schnittstellen des BS2000 und weitere, ausführliche Informationen zum Thema finden Sie im Handbuch „Dateien und Volumes größer 32 GB“ [19].

4 Makroaufrufe

ADDPLNK – Poolkettungsnamen definieren

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form) (siehe Seite 870)

Der Makro ADDPLNK ordnet einem ISAM-Pool für den Auftrag einen Poolkettungsnamen zu, der in eine Pooltabelle eingetragen wird. Dieser Poolkettungsname muss über FILE-Aufruf (Operanden LINK und POOLLNK) in die Task File Table eingebracht oder im FCB (Feld POOLLNK) angegeben werden. Beim OPEN wird geprüft, ob für die Datei ein Poolkettungsname eingetragen ist und ob für diesen ein ISAM-Pool existiert.

Format

Operation	Operanden
ADDPLNK	POOLNME=poolname ,LINKNME=name [,CATID=catid] [,SCOPE={TASK | USERID | USERGROUP | HOST}] MF=L,POOLNME=poolname
	MF=E,PARAM={adr | (r)}
	MF=D[,PREFIX=pre]
	MF=C[,PREFIX=pre][,MACID=macid]

Operandenbeschreibung

CATID = catid
gibt an, welchem Pubset der ISAM-Pool zugeordnet ist, und muss mit der CATID-Angabe im CREPOOL-Makro übereinstimmen.

Voreinstellung: Default-Catid des Auftrags

LINKNME = name
ordnet dem ISAM-Pool „poolname“ den Poolkettungsnamen „name“ zu. „name“ kann 1-8 Zeichen lang sein; erlaubter Zeichenvorrat: alle Buchstaben und Ziffern sowie die Sonderzeichen $, # und @ (entsprechend den Regeln zur Bildung von Dateinamen).

MACID
wird nur in Verbindung mit MF=C ausgewertet und legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: MACID = ISA

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870)

= adr
adr ist die symbolische Adresse (der Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

POOLNME = poolname
beschreibt den ISAM-Pool, der für die Dateiverarbeitung genutzt werden soll. „poolname“ ist der Name, mit dem der ISAM-Pool angelegt wurde (siehe Makro CREPOOL, Seite 241).

PREFIX
wird nur in Verbindung mit MF=C oder MF=D ausgewertet und legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein Zeichen langes Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen sollen.

SCOPE
gibt den Geltungsbereich des ISAM-Pools an; SCOPE muss denselben Wert haben wie im CREPOOL-Makro.

= TASK
Der Poolkettungsname wird dem tasklokalen ISAM-Pool „poolname“ zugeordnet.

= USERID
Dieser Geltungsbereich wird nur noch aus Kompatibilitätsgründen unterstützt (siehe Makro CREPOOL, Seite 241).

= USERGROUP
Dieser Geltungsbereich wird nur noch aus Kompatibilitätsgründen unterstützt (siehe Makro CREPOOL, Seite 241).

= HOST
Der Poolkettungsname wird dem taskübergreifenden ISAM-Pool „poolname“ zugeordnet.

Returncodes

Die mit der C- oder D-Form des Makros generierten Feldnamen und EQU-Anweisungen für die Returncodes beginnen standardmäßig mit der Zeichenfolge DISA, die durch PREFIX und MACID geändert werden kann.

Die Returncodes werden im Standardheader der Operandenliste hinterlegt.

Haupt-Returncode		**Bedeutung**
DISAOK	X'0000'	Makroaufruf war erfolgreich
DISANPAR	X'0001'	auf die Operandenliste kann nicht zugegriffen werden
DISANREM	X'0002'	der mit Catid bezeichnete Pubset befindet sich an einem Host-Rechner, auf dem eine BS2000-Version läuft, die ISAM-Pools nicht unterstützt
DISANCAT	X'0003'	die Katalogkennung „catid“ ist im System nicht bekannt
DISANACC	X'0004'	zum Pubset „catid“ besteht keine Verbindung
DISAINVN	X'0005'	der Poolname oder Poolkettungsname ist ungültig
DISANCLA	X'000A'	der Poolkettungsname wurde schon vergeben und die bereits bestehende Zuordnung kann nicht aufgehoben werden
DISASYSE	X'000B'	während der Makrobearbeitung trat ein interner Fehler auf
DISANOPL	X'000D'	es existiert kein Pool mit dem angegebenen Namen
DISARLNK	X'FFFF'	Makroaufruf konnte nicht ausgeführt werden: Sub-Returncode1 auswerten! (Linkage Fehler)

BTAM – Banddateien verarbeiten (Typ S)

Makrotyp: S-Typ (E-Form/L-Form, siehe Seite 870)

Alle Benutzeranforderungen für BTAM werden über diesen Makroaufruf abgewickelt. In den Operandenbeschreibungen wird für „Magnetbandkassette“ die Abkürzung MBK verwendet.

Format

Operation	Operanden
BTAM	fcbadr [,{RDWT | RBID | RD | RDBF | CHK | MINF | POS | REV | REVWT | RNT | RNTL | RT | RTL | SYNC | WRT | WRTWT | WT | ERG | BSF | BSR | FSF | FSR | REW | RUN | WTM}][,LEN=länge][,LOC={1 | 2 | relaus}]

(Teil 1 von 2)

Operation	Operanden
	[,PARMOD={24 | 31}][,REQNO=zahl] MF=L
	MF=({E,adr | E,(r)})

(Teil 2 von 2)

Operandenbeschreibung

fcbadr
gibt die Adresse des FCB an, mit dem die zu verarbeitenden Datei in Verbindung steht.

RDWT
liest Magnetband vorwärts und wartet das Ende der Operation ab, bevor die Steuerung dem Anwenderprogramm übergeben wird (Standardfunktion)

CHK
prüft, ob die vorausgegangene Ein-/Ausgabeoperation abgeschlossen ist. Ist die Ein-/Ausgabeoperation noch nicht abgeschlossen, wird die Steuerung an die LOC-Adresse übertragen. Andernfalls ist die Operation äquivalent zu WT.

MINF
holt eine Medium-Information bei der Bearbeitung von optischen Platten. Die Bedienung von optischen Platten im BS2000 erfolgt über eine MBK-Emulation.
Der Bereich, in den die Information geschrieben werden soll und seine Länge (derzeit 128 Byte) müssen über die Operanden LOC und LEN spezifiziert werden. Das Layout der Ausgabeinformation wird mit GC NDWMINF beschrieben.

POS
nur mit PARMOD=31: Band positionieren (siehe „Operationscodes“ auf Seite 122)

RBID
nur mit PARMOD=31: Bandposition bestimmen (siehe „Operationscodes“ auf Seite 122)

RD
liest Magnetband vorwärts

RDBF
Für Magnetbandkassetten, nur mit PARMOD=31: Daten blockweise aus dem Sicherstellungsbereich des MBK-Puffers in den Anwendungsbereich übertragen (siehe „Operationscodes“ auf Seite 122)

REV
liest Magnetband rückwärts

REVWT
liest Magnetband rückwärts und wartet das Ende der Operation ab, bevor die Steuerung dem Anwenderprogramm übertragen wird.

RNT
Lesen ohne Datentransfer, mit Meldung bei kürzerer Länge als erwartet

RNTL
Lesen ohne Datentransfer, ohne Meldung bei kürzerer Länge als erwartet

RT
Lesen mit Datentransfer, mit Meldung bei kürzerer Länge als erwartet

RTL
Lesen mit Datentransfer, ohne Meldung bei kürzerer Länge als erwartet

SYNC
nur mit PARMOD=31: synchronisieren und Bandposition bestimmen (siehe „Operationscodes“ auf Seite 122)

WRT
schreibt auf Magnetband

WRTWT
schreibt auf Magnetband und wartet das Ende der Operation ab, bevor die Steuerung dem Anwenderprogramm übertragen wird.

WT
wartet die Beendigung der vorausgegangenen Ein-/Ausgabeoperation ab. Die Steuerung wird erst nach Abschluss der Operation bzw. der notwendigen Fehlerbehandlungen an das Benutzerprogramm zurückgegeben.

ERG
erzeugt Blocklücke; bei Wiederholung wird das Band gelöscht!
Die mit ERG ausgelöste Operation ist eine Schreiboperation, die an Stelle eines Bit-Musters ein „Blocklücken-Muster“ in der vom Anwender angegebenen Länge erzeugt (bei einigen Magnetbandtypen ist diese Länge festgelegt).

BSF
Band um eine Abschnittsmarke zurücksetzen

BSR
Band um einen Block zurücksetzen

FSF
Band um eine Abschnittsmarke vorsetzen

FSR
Band um einen Block vorsetzen

REW
Rückspulen bis Bandanfang

RUN
Rückspulen und Band entladen, anschließend ist nur noch CLOSE möglich

WTM
Abschnittsmarke schreiben

LEN = länge
gibt die Länge der einzelnen Blöcke bzw. bei geketteter Ein-/Ausgabe (Operand CHAINIO in FILE/FCB) die Länge der Transporteinheit an.
Wird LEN nicht angegeben, errechnet sich die Länge einer Transporteinheit aus dem Produkt von „Blockgröße“ und Kettungsfaktor, wobei die Blockgröße bei RECFORM=U durch den Inhalt des unter RECSIZE angegebenen Registers bestimmt ist und bei RECFORM=F durch BLKSIZE.

Ohne Kettung holt sich das System die Angaben nur aus dem RECSIZE-Operanden beim Schreiben mit RECFORM=U, sonst aus dem BLKSIZE-Operanden.

Eine Längenangabe mit LEN darf bei RECFORM=U so ausgelegt sein, dass der letzte Block innerhalb einer Transporteinheit kürzer als die vorhergehende ist. Bei RECFORM=F sollte die angegebene Länge ein Vielfaches der gegebenen BLKSIZE betragen (falls dem nicht so ist, wird der Auftrag aber nicht abgewiesen).
Die zu lesende Länge wird immer und nur der aktuellen Längenangabe entnommen. Die tatsächliche Blocklänge steht nach erfolgreichem Lesen im RECSIZE-Register.
Im Zusammenhang mit dem Operationscode RDBF gibt LEN an, wie lang die aus dem Puffer zu sichernden Blöcke sind. Gilt RECFORM=U/V, wird die aktuelle Blocklänge im RECSIZE-Register zurückgemeldet.

LOC
gibt den Bereich an, in den gelesen oder aus dem geschrieben werden soll.

Voreinstellung: IOAREA1 – bei der ersten Ein-/Ausgabe
– bei Wechsel von LOC=relaus auf eine IOAREA
– wenn zuletzt IOAREA2 verwendet wurde
– wenn IOAREA2 nicht definiert ist
IOAREA2 – falls zuletzt IOAREA1 verwendet wurde

= relaus
Bereichsadresse im Makroaufruf.
Wird eine CHK-Operation verlangt, so muss der LOC-Operand die Form LOC=relaus aufweisen. Die Steuerung wird an die mit LOC definierte Adresse übertragen, wenn die überprüfte Operation noch nicht abgeschlossen war. Der adressierte Bereich muss nicht mit IOAREA1/2 zusammenfallen.

= 1
verweist auf die IOAREA1-Adresse im FCB

= 2
verweist auf die IOAREA2-Adresse im FCB
Im TU-FCB wird angezeigt, welcher Ein-/Ausgabebereich genutzt wurde:
– Im 31-Bit-TU-FCB im Feld ID1BLWB
– Im 24-Bit-TU-FCB im Feld ID1LWB
Im Zusammenhang mit den Operationscodes POS, RBID, RDBF und SYNC hat LOC folgende Funktion:
– POS: Adresse der Positionsangabe (9 Byte), auf die positioniert wird
– RBID: Adresse, an der die 9 Byte lange Positionsangabe ausgegeben wird
– RDBF: Adresse des Bereichs, in den der sichergestellte Block gebracht wird
– SYNC: Adresse, an der die 9 Byte lange Positionsangabe ausgegeben wird
Auch bei diesen Operationscodes wird der Pointer im TU-FCB (ID1BLWB bzw. ID1LWB) auf den zuletzt genutzten Bereich gesetzt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang, Seite 870 beschrieben.

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.

Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst. Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

REQNO = zahl
zahl ≤ 8, „zahl“ bestimmt die Nummer der Ein-/Ausgabeanforderung bzw. des zugehörigen Aufrufs. Durch verschiedene Nummern gekennzeichnet, können mehrere asynchrone Lese- oder Schreiboperationen veranlasst werden. Jede dieser Lese-/Schreiboperationen muss mit der Operation WT (versehen mit der entsprechenden Nummer) abgeschlossen werden. Die maximale Anzahl gleichzeitiger Ein-/Ausgabeoperationen wird im FCB, Operand BTAMRQS, festgelegt.

Operationscodes

POS – Band auf eine bestimmte Stelle positionieren

Zum Wiederaufsetzen, z.B. nach Schreibfehlern mit Datenverlust: der Anwender kann die Bandposition, die er durch eine frühere RBID-Operation erhalten hat, an der mit LOC bestimmten Adresse angeben, das Band wird dann entsprechend positioniert.

RBID – aktuelle Bandposition (Blocknummer) bestimmen

Jeder Block auf einem Magnetband, auch eine Abschnittsmarke, ist durch seine Bandposition identifizierbar.
Bei herkömmlichen Magnetbändern erhält der Anwender die Bandposition als Wertepaar (TM- und Record-Counter) zurück. Sie wird dem Anwender nach einem RBID-Befehl (oder SYNC-Befehl, siehe „SYNC – Synchronisieren und Markierungspunkte setzen“ auf Seite 124) an der Adresse angezeigt, die durch den Operanden LOC definiert ist (siehe Operandenbeschreibung) oder – wenn LOC nicht angegeben wurde – in einem der im FCB definierten Ein-/Ausgabebereiche. Die Positionsangabe (die ersten acht Byte) ist abhängig

vom Verarbeitungszustand und liefert im Fehlerfall die in der Tabelle auf Seite 123 aufgeführten Informationen im FCB-Feld ID1ECB. Das 9. Byte gibt an, wie die Postitionsangabe zu interpretieren ist:

2**7 = 1: TM-/Record-Counter

2**6 = 1: Block-ID

2**0 = 1: keine gültige Positionsangabe

Vor Absetzen eines RBID-Befehls müssen alle noch ausstehenden asynchronen Lese-/Schreiboperationen (im MAV-Modus) mit einem WAIT abgeschlossen werden; bei synchroner Dateiverarbeitung führt BTAM den WAIT automatisch durch.

Bei Magnetbandkassetten wird die Bandposition als „Blocknummer“ (oder Block-ID) bestimmt.

Rückinformationen

Ereignis	**Rückinformation**	**Bedeutung**	**Maßnahme**
erfolgreiche Ausführung	Blocknummer	zeigt die Position des Datenblocks an, der als nächster in den MBK-Puffer geschrieben oder aus dem Puffer gelesen wird	----
nicht erfolgreiche Ausführung	Blocknummer +Fehler DC7C	Bandposition wurde gesichert; die Blocknummer zeigt den letzten fehlerfrei geschriebenen Block	neu positionieren mit POS
	Blocknummer undefiniert + Fehler DC7B	Bandposition konnte nicht gesichert werden	Rücksetzen auf einen Wiederaufsetzpunkt oder Programmabbruch
	Blocknummer undefiniert + Fehler DC79	Bandposition konnte nicht gesichert werden	Programmabbruch
	Blocknummer + Fehler DC77	Ein-/Ausgabefehler; die Bandposition wird so zurückgegeben, wie sie nach der Fehlerbehandlung vorliegt	----
	Blocknummer undefiniert + Fehler DC77	Ein-/Ausgabefehler mit „Positionsverlust“	Rücksetzen auf einen Wiederaufsetzpunkt oder Programmabbruch

RDBF – nur für MBK – bei nicht behebbarem Schreibfehler die sichergestellten Daten des MBK-Puffers in den Anwenderbereich übernehmen

Bei gepuffertem Schreiben kann ein Fehler für Datenblöcke auftreten, für die der Anwender bereits eine positive Quittung erhalten hat. Die Geräteverwaltung versucht, die dem Kassettenpuffer bereits übergebenen Daten und die Bandposition zu sichern, sodass der Anwender eine ordnungsgemäße Fehlerbehandlung durchführen kann. Er kann den fehlerhaften Block und die darauf folgenden ebenfalls bereits positiv quittierten, aber noch nicht auf das Band geschriebenen Blöcke dem Sicherstellungsbereich des Puffers entnehmen und erneut bearbeiten, z.B. auf einen anderen Datenträger ausgeben.

Bevor die Datei geschlossen wird (CLOSE) oder Bandwechsel mit FEOV eingeleitet wird, muss der Anwender den Datenträger mit POS hinter den letzten erfolgreich auf das Band geschriebenen Block positionieren. Das „Herauslesen“ der Blöcke erfolgt nach dem Prinzip „last-in – first-out“, die Anzahl der gespeicherten Blöcke ist im Feld ID1BLANZ des TU-FCB enthalten. Für jeden Block muss ein RDBF-Befehl gegeben werden, der Eingabebereich wird wie bei einem normalen Leseaufruf versorgt: die Adresse ist über LOC im BTAM-Makroaufruf oder durch IOAREA im FCB gegeben, die Länge durch LEN im BTAM-Makroaufruf oder durch FCB-Angaben, analog dem Lesen von Band. Die Blöcke werden nur in der angegebenen Länge übertragen. Ist für die Datei RECFORM=V/U definiert, wird die echte Länge des gesicherten Blocks in dem Register angezeigt, das durch RECSIZE in FILE oder FCB definiert ist.

SYNC – Synchronisieren und Markierungspunkte setzen

Die im MBK-Puffer enthaltenen Daten werden auf Band geschrieben. Bei synchroner Verarbeitung (nicht MAV-Modus) veranlasst BTAM einen evtl. noch ausstehenden WAIT; bei asynchroner Verarbeitung (MAV-Modus) muss der Anwender dafür sorgen, dass vor Absetzen des SYNC-Befehls alle ausstehenden WAITS ausgeführt werden. Implizit wird mit dem SYNC-Aufruf auch ein RBID-Aufruf veranlasst, d.h. dem Anwender wird im Ein-/ Ausgabebereich die aktuelle Bandposition mitgeteilt. Der Anwender kann dieses Verhalten nutzen, um Fixpunkte zu setzen, z.B. im Hinblick auf evtl. später auftretende Fehler: er kann auf diesen Punkten wieder aufsetzen, einen Bandwechsel veranlassen und auf dem Folgeband die Verarbeitung fortsetzen.

Hinweise zur Programmierung

1. Der BTAM-Makroaufruf zerstört die Register 0, 1, 14 und 15.
2. Missbrauch von FSF und FSR kann dazu führen, dass das Band bis zum Ende durchläuft und vom Operator wieder zurückgespult werden muss (off-line).
3. Bei jedem nicht erfolgreichen Verzweigen nach BTAM wird die Steuerung an die Adresse abgegeben, die im EXIT-Operanden des FCB definiert ist. Im FCB wird ein entsprechender sedezimaler Fehlerschlüssel abgesetzt.

4. Gibt ein Anwender Blöcke fester Länge an (Format F), liest jedoch Blöcke anderer Länge als angegeben, wird die Steuerung an den Ausgang ERRADDR im EXLST-Makroaufruf übertragen.

   Überschreitet die Länge des Satzes die definierte Länge, so wird im „Ablaufteil-Markierungsbyte“ das „abnormale Beendigungs-Bit“ und im Fehler-Byte das „Satzlängen-Fehler-Bit“ gesetzt, sofern es sich nicht um eine RTL- bzw. RNTL-Operation unter gleichzeitiger Verwendung von Kettung und/oder MAV-Modus handelt (in diesem Fall erfolgt keine Benachrichtung des Benutzers).

   Ist der Block jedoch kleiner als die definierte Länge, so wird das „abnormale Beendigungs-Bit“ gesetzt. Es wird jedoch kein Fehlerbyte übermittelt (die Restlänge ist in den Sense-Byte 2 und 3 gespeichert; siehe Makro NDWERINF, Seite 750). Eine Benachrichtigung des Benutzers erfolgt nicht, wenn es sich um eine RTL- bzw. RNTL-Operation handelt.

5. Gibt ein Anwender Blöcke variabler Länge an (Format U bzw. V) oder verwendet er die Operationcodes RTL bzw. RNTL, liest jedoch Blöcke mit größerer Länge als angegeben, so wird die Steuerung an den Ausgang ERRADDR im EXLST-Makroaufruf dann übertragen, wenn ohne Kettung und ohne MAV-Modus gearbeitet wird. Bei den Operationen RT oder RNT wird generell der Ausgang ERRADDR aktiviert. In allen anderen Fällen erfolgt bei Format U/V keine Information des Benutzers.

   Zusammengefasst ergibt sich also folgendes Bild bzgl. der Benachrichtigung des Benutzers.

Operation	**Format**			
	RECFORM=F		**RECFORM=U/V**	
	Block < gg. Länge	**Block > gg. Länge**	**Block < gg. Länge**	**Block > gg. Länge**
RD(WT)	ja	ja	nein	bedingt *)
R(N)T	ja	ja	ja	ja
R(W)TL	nein	bedingt *)	nein	bedingt *)

   *) Benachrichtigung falls nicht Kettung und nicht MAV-Modus

6. Der Anwender braucht, falls nicht im MAV-Modus gearbeitet wird, explizit keine „WAITs“ anzugeben. BTAM gibt automatisch vor jeder neuen Operation einen WAIT aus. Tritt jedoch in diesem WAIT ein Fehler auf, so wird der Fehlerschlüssel X'0C77' ausgegeben. Die neue Operation wird nicht ausgeführt.

   Es ist aber zu beachten, dass dieser implizite WAIT-Aufruf nur bei Arbeiten mit mehreren Ein-/Ausgabebereichen sinnvoll ist. Da die Daten im Ausgabebereich beim Schreiben grundsätzlich bis zum erfolgten WAIT erhalten bleiben müssen bzw. beim Lesen

der Eingabebereich erst nach erfolgtem WAIT ordnungsgemäß gefüllt ist, muss im Falle der Verwendung nur eines Ein-/Ausgabebereiches der Anwender **selbst** für das Absetzen eines WAITs sorgen, sobald er seinen einzigen Bereich erneut verwenden will.

7. BTAM verwendet nicht den EXLST-Ausgang EOFADDR (Dateiende). Wird von einem RD, RDWT, RNT, RNTL, RT, RTL, REV oder REVWT eine Abschnittsmarke gelesen, wird die Steuerung an die ERRADDR-Adresse übertragen.

   Das Programm kann nach dem WAIT die 5 Status-Byte im FCB (SDB, FB1, FB2, FB3, AMB) auswerten. Die Steuerung wird ebenfalls der ERRADR-Adresse übertragen, falls versucht wird, ein auf Bandanfang positioniertes Band rückwärts zu lesen.

8. Bei den Operationen REV und REVWT ist die Adresse für das erste Byte, in das Daten gelesen werden, definiert durch:

   LOC + LEN - 1.

9. Bei einer Datei, die mit BTAM eröffnet wurde, kann der SAM-Makroaufruf FEOV verwendet werden. Bei Erkennen des Dateiendes auf dem aktuellen Band, verzweigt BTAM zum sonst nicht verwendeten Ausgang EOFADDR.

10. CHAINIO mit Magnetbandkassetten: die Transporteinheit sollte nicht größer sein als die Puffergröße, und die Puffergröße sollte als n*BLKSIZE darstellbar sein. Wird im Fehlerfall die Anzahl noch im Puffer befindlicher Blöcke angezeigt (TU-FCB: ID1BLANZ), bezieht sich diese immer auf einzelne Blöcke, nicht auf Transporteinheiten; auch die RDBF-Operationen betreffen immer nur einzelne Blöcke.

11. MAV-Modus mit Magnetbandkassetten: für BTAMRQS im FCB können kleinere Werte gewählt werden als bei Magnetbändern, da die Benutzer-Ein-/Ausgabebereiche schneller wieder frei werden.

12. Im Fehlerfall kann es vorkommen, dass nicht der gesamte Inhalt des MBK-Puffers auf die Magnetbandkassette geschrieben werden konnte. Auf die vom Betriebssystem aus dem MBK-Puffer in einen Sicherstellungsbereich übernommenen Blöcke kann der Anwender wieder Bezug nehmen, was einem Lesen von Band in umgekehrter Richtung entspricht. Anschließend sollte der Anwender versuchen, die Magnetbandkassette hinter den letzten ordnungsgemäß geschriebenen Block zu positionieren.

    Der Anwender muss dafür sorgen, dass vor weiteren Schreibaufrufen (speziell vor dem Schreiben der EOV-/EOF-Kennsätze) das Magnetband eine definierte Position hat (POS-Befehl); andernfalls können Daten überschrieben werden. RBID nach Fehler DC7C liefert die Position nach dem letzten korrekt auf Band geschriebenen Block.

    Der Anwender erreicht einen ordnungsgemäßen Abschluss des Bandes mit Schreiben von Endekennsätzen, wenn er anschließend einige Blöcke rückwärts liest oder zurückpositioniert (falls er auf die bereits geschriebenen Daten verzichten kann).

13. Wurden bei einem Schreibfehler nicht alle Blöcke des Kassettenpuffers auf das Band geschrieben, im Sicherstellungsbereich des Puffers aber noch gesichert, muss als Nächstes die Operation RDBF oder RBID veranlasst werden, als Zeichen dafür, dass der Anwender auf die Daten des MBK-Puffers nicht verzichtet. Diese Daten stehen ihm sonst später nicht zur Verfügung. Die Operationen RDBF und RBID können in beliebiger Reihenfolge abgesetzt werden; der Pufferinhalt bleibt solange erhalten, wie keine andere Operation veranlasst wird. Ausgenommen sind dabei noch ausstehende WAITs im MAV-Mode, die vom Benutzer vor Abgabe des RDBF- und RBID-Befehls aufgerufen werden dürfen (und vor dem RBID auch müssen); naturgemäß werden dann alle den Fehler DC7C melden.

    Verzichtet der Benutzer nach der Meldung DC7C auf eine anschließende Positionierung, so geht die Bandposition auf „UNDEFINED“ über. Dieser Zustand „UNDEFINED“ kann nur durch Positionierung (auch REW und UNL) aufgehoben werden.

14. Beim Kennzeichnen von markanten Punkten (Fixpunkte) durch Absetzen von SYNC-Befehlen wird verlangt, dass vorher alle angestoßenen Ein-/Ausgaben mit einem WAIT abgeschlossen wurden.

15. Im MAV-Modus müssen bei einer Aufruffolge, wo „Schreiben“ nach einem „Lesen rückwärts“ erfolgt, vor dem „Schreiben“ alle ausstehenden WAITS abgesetzt sein.

16. Es ist eine Eigenart der Magnetbandkassetten, dass ein bei Ausgabeoperationen auftretender Fehler erst zum Zeitpunkt des Transfers vom Puffer auf das Band gemeldet wird; der Benutzer erfährt von einem aufgetretenen Fehler somit u.U. erst zum Zeitpunkt eines späteren Auftrags. Es kann also vorkommen, dass der Anwender die Meldung „Bandende“ noch vor der Meldung eines davor aufgetretenen Fehlers bekommt. Die Meldung „Bandende“ bedeutet, dass ab dem Zeitpunkt der Meldung keine weiteren Schreibaufträge vom Benutzer abgesetzt werden sollen, damit gewährleistet ist, dass noch alle im Puffer befindlichen Daten und die Endekennsätze auf das Band passen. Die Meldung „Bandende“ ist bei Magnetbandkassette nicht vom Erkennen einer Endemarke auf dem Band abhängig, sondern wird bereits beim Transfer der Daten in den Puffer, abhängig von dessen Füllgrad und anderen Kriterien, ausgegeben.

17. Bei den Makroaufrufen FEOV oder CLOSE für eine BTAM-Datei, die INOUT eröffnet wurde, wird die Datei als Ausgabedatei behandelt, wenn für die Datei im Laufe der Verarbeitung eine WRITE-mit-WAIT-Operation angefordert wurde (WRT, WRTWT, WTM). Wurde keine WRITE-mit-WAIT-Operation angefordert, wird die Datei als Eingabedatei behandelt.

18. BTAM gibt den Fehlerschlüssel X'0C95' aus
    - wenn ein ungültiger Gerätetyp angegeben wurde
    - wenn der Operationscode ungültig ist
    - wenn im FCB-Operanden RECSIZE eines der Register 0, 1, 13, 14 oder 15 angegeben wurde
    - wenn der Wert des Operanden LEN im BTAM-Makroaufruf ≤ 0 ist oder wenn er, falls LOC=relaus nicht angegeben ist, größer als BLKSIZE ist (dann nämlich wird IOAREA1/2 verwendet, die die Größe von BLKSIZE hat)
    - wenn bei einer OUTPUT-eröffneten Datei eine Leseoperation angefordert wurde
    - wenn im MAV-Modus die REQNO-Angabe größer ist als maximal erlaubt
    - wenn im MAV-Modus vor einem Lese-/Schreibaufruf für die gleiche REQNO der WAIT-Aufruf fehlt
    - wenn nach RUN ein BTAM-Aufruf gegeben wurde (es ist nur CLOSE erlaubt) oder wenn nach einem nicht erfolgreichen FEOV ein BTAM-Aufruf gegeben wird.
    - wenn für eine nicht für BTAM eröffnete Datei BTAM-Aufrufe gegeben werden (außer in EXLST-Routinen zur Kennsatzbehandlung)
19. Beim Erweitern einer Banddatei sollte ein unmittelbarer Wechsel von Lese- und Schreibbefehl vermieden werden. Wegen fehlender Löschüberlappung kann das bei einzelnen Magnetbandgeräten zu Schwierigkeiten führen.

    Eine Folge, die immer korrektes Verhalten garantiert:

BTAM-Makroaufruf	**Erklärung**
BSR, FSR, ...	Band vor Block n positionieren
RD	Lesen (Retten) von Block n
BSR	Band wieder vor Block n positionieren
WRT	Block n erneut auf das Band schreiben
WRT/WTM	Block n+1 (oder Bandmarke) schreiben

20. In bestimmten Fällen wird beim Lesen von Daten mit Kettung eine falsche Blocklänge nicht angezeigt (siehe Seite 125).

    Die Kette für die Eingabe CCWs läuft weiter. Die Bereiche, wohin die Eingaben der einzelnen Blöcke erfolgen, werden aber in dem der normalen Blockgröße entsprechenden Rhythmus weitergeschaltet, d.h. es werden in den Fällen, wo Blöcke kürzerer Länge auftreten, in den Ein-/Ausgabe-Bereichen Lücken vorhanden sein.

21. Eine Angabe über verwendete Puffer erfolgt bei PARMOD=24 im Feld ID1LWB des P1FCB bzw. bei PARMOD=31 im Feld ID1BLWB bei der Durchführung eines WAITs in folgender Weise (asynchrone Verarbeitung): Bei Eingabe-Operationen wird der Puffer vermerkt, für den die letzte WAIT-Operation ohne Fehler-Rückmeldung erfolgt ist. Bei Ausgabe-Operationen ist der Puffer angegeben, der für die letzte Ausgabe (die noch nicht unbedingt beendet ist) verwendet wurde.

    Bei synchroner Verarbeitung gilt: bei PARMOD=24 wird im FCB-Feld ID1LWB der Puffer angegeben, der als Letzter verwendet und für den ein erfolgreicher WAIT abgesetzt wurde, bei PARMOD=31 wird das FCB-Feld ID1BLWB verwendet.

22. Bei der Festlegung der Block- und Transporteinheits-Größen durch den Benutzer ist auf einen besonderen Aspekt hinzuweisen: Hinter jeder Ein-/Ausgabe-Operation durch den Gerätetreiber steht ein Fixieren der betroffenen Seiten.
    Bei gekettetem Ein-/Ausgabe-Betrieb wird das Fixieren bei jedem CCW der Kette veranlasst. Dies aber ist vom Memory Management aus nur bis zu einer Anzahl von 63 Fixierungen pro Seite möglich; d.h., wenn ein Auftrag so abgegeben wird, dass u.U. mehr als 63 CCWs, die sich auf dieselbe Seite beziehen, aufgebaut werden (z.B. bei Verwendung kleiner Blockgrößen und großer Transporteinheit-Längen), kann es zu einem CSTAT-Fehler kommen.

23. Bei Bändern mit Standard-Kennsätzen überprüft BTAM vor Schreibbefehlen, ob die aktuelle Bandposition ein Schreiben dorthin erlaubt. Ist dies nicht der Fall, so wird der Auftrag mit USERERR und Fehlercode 0C9D abgewiesen.

Returncodes

Im TU-FCB werden die fünf Status-Byte (SDB, FB1, FB2, FB3, AMB) abgelegt. Zusätzlich werden im Feld ID1LRCRB noch folgende Informationen abgesetzt:

Byte 1 X'04' Bandende/Bandanfang erreicht
X'02' Block kürzer als BLKSIZE-Wert (Blocklänge)
X'05' Block länger als BLKSIZE-Wert (Blocklänge)
X'01' Bandmarke erkannt
X'08' undefinierter Fehler (unrecoverable)
X'09' Parity Fehler (inoperabel)
X'0C' Gerät defekt
X'0D' Gerät in Betrieb
X'0E' Folge-Fehler auf Magnetbandkassette; Position nach defektem Block
X'0F' Bandformat und Gerätetyp sind nicht kompatibel

Byte 2 Request-Nummer (nötig, da EXIT ERRADR nur einmal pro FCB)

Byte 3/4 Blocknummer innerhalb einer geketteten Ein-/Ausgabe-Anforderung, bei der ein Fehler aufgetreten ist, bzw. Nummer des Blockes, mit dem das Bandende erreicht wurde (dieser Block wird noch geschrieben).

Zum Abfragen der im ersten Byte des Feldes ID1LRCRB abgesetzten Werte steht eine DSECT (Makroaufruf: DLRC) zur Verfügung.

Bei Arbeiten im asynchronen Modus (Operand REQNO) werden bei Auftreten von Fehlern bei einer Ein-/Ausgabe die bereits angenommenen weiteren Ein-/Ausgabe-Anforderungen nicht mehr gestartet, sondern nach dem vom Benutzer abgesetzten WAIT logisch beendet. In einem solchen Fall ist neben der für die fehlerhafte Ein-/Ausgabe gültigen Fehlerinformation die Blocknummer in den Byte 3 und 4 des Feldes ID1LRCLB mit dem Wert 1 besetzt, und der Wert im RECSIZE-Register (bei RECFORM=U) ist 0.

Es ist gewährleistet, dass der Benutzer jederzeit, auch vor Abgabe aller noch ausstehenden WAITs, einen CLOSE-Aufruf geben kann. Alle zu diesem Zeitpunkt evtl. noch anstehenden Ein-/Ausgabe-Anforderungen werden dabei logisch beendet. Es sind aber noch nicht unbedingt alle Anforderungen durchgeführt.

CATAL – Katalogeintrag bearbeiten

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Der CATAL-Makroaufruf erstellt oder ändert Katalogeinträge. Insbesondere kann der Anwender damit Merkmale für Datei- und Datenschutz, sowie den gültigen Zeichensatz und Performance-Attribute festlegen; er kann auch temporäre Dateien in permanente umwandeln oder umgekehrt.

Sollen Merkmale bereits bestehender Katalogeinträge geändert werden, muss der Operand STATE=*UPDATE angegeben werden. Es werden allerdings nur die Dateimerkmale, d.h. die Felder des Katalogeintrags, geändert, deren zugehörige Operanden mit gültigem Operandenwert angegeben werden.

Ein Katalogeintrag kann nur geändert werden, wenn Schreibzugriff nicht durch ein Kennwort unterbunden ist. Dieses muss dann vor Zugriff auf den Katalogeintrag mit dem Kommando ADD-PASSWORD (siehe Handbuch „DVS Einführung“ [1]) in die Kennworttabelle des Auftrags eingetragen werden.

Mit CATAL können Dateien, Dateigenerationen und Dateigenerationsgruppen katalogisiert werden. Für Dateien und Dateigenerationsgruppen lassen sich die Schutzmerkmale verändern, die Schutzmerkmale von Dateigenerationen sind durch den Gruppeneintrag festgelegt.

Der Makro CATAL unterstützt die Funktion „Default-Protection“.

Die Verschlüsselungsmerkmale einer Datei können mit dem Makro CATAL nicht geändert werden.

Temporäre Dateien

Da temporäre Dateien auftragsabhängig sind, kann für sie kein Dateischutz vereinbart werden, d.h. die entsprechenden Standardmerkmale können nicht verändert werden. Beim Einrichten einer temporären Datei bzw. wenn eine permanente Datei in eine temporäre umkatalogisiert wird (oder umgekehrt), sind nachfolgende Punkte zu beachten.

Temporäre Dateien können von einem nichtprivilegierten Benutzer nur auf dem Default-Pubset seiner Benutzerkennung angelegt werden.

– Einrichten einer temporären Datei:

  Die temporäre Datei bekommt folgende Attribute (eine explizite Angabe anderer Werte ist hierfür generell nicht zulässig):

EXPIRATION-DATE	= <tagesdatum>	BACKUP-CLASS	= *E
USER-ACCESS	= *OWNER-ONLY	READ-PASSWORD	= *NONE
WRITE-PASSWORD	= *NONE	EXEC-PASSWORD	= *NONE
ACL	= *NO	GUARDS	= *NONE
BASIC-ACL	= *NONE	ACCESS	= *WRITE
FREE-FOR-DELETION	= *NONE	AVAILABILITY	= *STD
MANAGEMENT-CLASS	= *NONE		

  Das Attribut DISK-WRITE wird standardmäßig auf *BY-CLOSE gesetzt, kann jedoch durch explizite Angabe auf *IMMEDIATE gesetzt werden. Ebenso kann für das Attribut MIGRATE mit dem Standardwert *INHIBITED durch explizite Angabe der Wert *FORBIDDEN gesetzt werden.

– Umkatalogisieren von temporär nach permanent:

  Wenn keine expliziten Angaben gemacht werden, bekommt die permanente Datei folgende Attribute:

BACKUP-CLASS	= Wert der Class2-Option BACKUP
DISK-WRITE	= *IMMEDIATE
MIGRATE	Wenn für die temporäre Datei MIGRATE=*INHIBITED vereinbart war, wird für die permanente Datei *ALLOWED gesetzt (MIGRATE=*FORBIDDEN bleibt unverändert).

  Die übrigen Attribute werden unverändert von der temporären Datei übernommen.

– Umkatalogisieren von permanent nach temporär:

  Die temporäre Datei bekommt die gleichen Attribute wie beim Einrichten einer temporären Datei, siehe oben.

  Der einzige Unterschied betrifft den Wert für MIGRATE: Wenn für die permanente Datei MIGRATE=*ALLOWED vereinbart war, wird für die temporäre Datei *INHIBITED gesetzt (MIGRATE=*FORBIDDEN bzw. *INHIBITED bleibt unverändert).

– In einem SM-Pubset wird das Umbenennen von temporär nach permanent und umgekehrt abgewiesen, wenn eine gleichzeitige Änderung der Dateiattribute ein Umallokieren auf einen anderen Volume-Set (S0-Migration) erfordert.

– Das Umkatalogisieren einer Arbeitsdatei (WORK-FILE) oder einer Datei auf einem Net-Storage-Volume in eine temporäre Datei oder umgekehrt ist nicht möglich.

Dateigenerationsgruppen (FGG)

Beim Erstellen oder beim Zugriff auf den Katalogeintrag von Dateigenerationsgruppen sind einige Besonderheiten zu beachten:

– Will der Anwender mit einer Dateigenerationsgruppe arbeiten, muss er den Gruppeneintrag erstellen, bevor er die erste Generation katalogisiert. Im Gegensatz zu Dateien und Dateigenerationen, die auch mit FILE katalogisiert werden können, kann der Gruppeneintrag auf Programmebene nur mit dem CATAL-Makroaufruf erstellt werden.

– Wenn der Index der Dateigenerationsgruppe auf öffentlichem Datenträger eingerichtet wird (keine Angabe von VOLUME und/oder DEVICE), können die Generationen sowohl auf öffentlichem Datenträger als auch auf Bändern (Programmschnittstelle FILE) angelegt werden.
  Wird der Index der Dateigenerationsgruppe auf Privatplatte eingerichtet (Angabe von VOLUME/DEVICE), dann können den Generationen ebenfalls nur auf Privatplatten (Programmschnittstelle FILE) angelegt werden.

– Dateien können in Dateigenerationen umkatalogisiert werden, wenn die Dateigenerationen noch nicht vorhanden sind. Eine Datei auf einem Net-Storage-Volume kann nicht in eine Dateigeneration umbenannt werden. Dateigenerationen können nicht in Dateien umkatalogisiert werden.

– Für einzelne Dateigenerationen einer Dateigenerationsgruppe können die Attribute (Operanden) STOCLAS, IOPERF, IOUSAGE, DISKWR und S0MIGR vergeben bzw. verändert werden.
  Die Einträge der Benutzer- bzw. Systemverwaltungs-Metainformation (Operanden USRINFO und ADMINFO) können für den Index der Dateigenerationsgruppe und jede einzelne Dateigeneration getrennt festgelegt werden.
  Die übrigen Attribute können nur für die ganze Dateigenerationsgruppe festgelegt werden. Sie werden automatisch vom Index auf alle katalogisierten Generationen vererbt.

- Das Attribut USER-ACCESS darf bei Dateigenerationsgruppen nicht auf SPECIAL gesetzt werden.
- Die Vergabe von Ausführungsrechten ist für Dateigenerationsgruppen nicht möglich und auch nicht sinvoll, da Generationen generell nicht ausgeführt werden können (Kommando CALL-PROCEDURE bzw. START-PROGRAM wird abgewiesen).
- Die Schutzattribute READ-PASSWORD, WRITE-PASSWORD und EXPIRATION-DATE schützen nicht den Index einer Dateigenerationsgruppe vor dem Anlegen neuer Generationen mit CATAL. Es können also davon unabhängig neue Generationen katalogisiert und dementsprechend in Abhängigkeit von der gewählten OVERFLOW-OPTION auch alte Generationen gelöscht werden.
- In eine Dateigenerationsgruppe mit dem Attribut ACCESS=READ kann mit CATAL <generationsname>,STATE=*NEW keine neue Generation angelegt werden, aber mit CATAL <datei>,<generationsname>,STATE=*UPDATE.
- Die Schutzattribute BASIC-ACL und GUARDS schützen sowohl die Dateigenerationsgruppe (Index) also auch jede einzelne Dateigeneration. In einer mit diesen Attributen schreibgeschützten Dateigenerationsgruppe kann ein Aufrufer ohne Schreibrecht also keine neue Dateigeneration anlegen.
- Dateigenerationsgruppen, die auf privaten Datenträgern gespeichert sind und für die kein Katalogeintrag existiert, werden als FOREIGN-Dateigenerationsgruppen bezeichnet. Sollen solche FGG wieder katalogisiert werden, muss zunächst der Gruppeneintrag erstellt werden. Für Dateigenerationsgruppen auf Privatplatte kann dazu der Operand STATE=*FOREIGN angegeben werden, wenn der Gruppeneintrag im F1-Kennsatz der Platte enthalten ist. Das System erstellt den Katalogeintrag dann aus dem F1-Kennsatz der über DEVICE- und VOLUME-Operand bezeichneten Privatplatte. Anschließend müssen die zugehörigen Dateigenerationen importiert werden (z.B. Makro FILE, Operand STATE=*FOREIGN).
- Soll eine Dateigenerationsgruppe importiert werden, deren Generationen auf Band oder auf Privatplatte gespeichert sind, deren F1-Kennsatz den Gruppeneintrag nicht enthält, müssen im CATAL-Makroaufruf zur Rekonstruktion des Gruppeneintrages die Operanden FIRST und mindestens einer der Operanden BASE oder LAST angegeben werden.
- Beim Einrichten einer Dateigenerationsgruppe im SM-Pubset muss implizit über die Vergabe einer Default-Storage-Klasse oder explizit durch Angabe des Operanden WORKGRP festgelegt werden, ob es sich um eine Gruppe von permanenten Generationen oder um eine Arbeits-Generationsgruppe handelt (Attribut WORK-FILE). Eine nachträgliche Änderung des Attributes ist nicht möglich.
  Wenn den zugehörigen Generationen eine Storage-Klasse zugewiesen wird (bei Erst-Allokierung über Programmschnittstelle FILE) oder die Storage-Klasse ausgetauscht wird, muss das WORK-FILE-Attribut in der Storage-Klasse mit dem Attribut der Gruppe übereinstimmen.

Dateien auf Magnetband/Magnetbandkassetten

Beim Erstellen oder beim Zugriff auf den Katalogeintrag von Banddateien sind Besonderheiten zu beachten, die sich durch das Speichermedium ergeben.

– Angaben zu Mehrbenutzbarkeit (SHARE), Zugriffsart (ACCESS) und Kennworte werden für Dateien mit Standardkennsätzen zum Zeitpunkt der Dateierstellung vom Katalogeintrag in die Dateikennsätze übertragen. Für FOREIGN-Dateien werden bei der Dateieröffnung die Angaben zu den Zugriffsrechten aus den Kennsätzen in den Katalogeintrag übernommen.

– Da Dateikennsätze auf einem Band nicht geändert werden können, ohne die Datei zu zerstören (Hardware-Einschränkung), und der Inhalt des Katalogeintrags einer Datei mit dem Inhalt der Dateikennsätze übereinstimmen muss, können Zugriffsrechte und Freigabedatum mit CATAL nicht mehr geändert werden, wenn die Banddatei einmal ordnungsgemäß eröffnet und geschlossen wurde.

– Wurde eine Banddatei mit einem FILE-Makroaufruf katalogisiert, können vor dem ersten Eröffnen der Datei die Dateischutzmerkmale mit CATAL verändert werden. Diese Merkmale werden dann bei der Dateierstellung ohne Prüfung in die Kennsätze übertragen. Auf diese Weise kann Schreibschutz (ACCESS=READ) für eine Datei vereinbart werden, die noch erstellt werden muss. Die Datei kann danach als Ausgabedatei eröffnet und erstellt werden; anschließend wird der Schreibschutz wirksam.

  Hinweis

  Wurde eine Banddatei mit FILE katalogisiert, ist sie mehrbenutzbar, wenn nicht mit CATAL vor dem ersten Eröffnen der Datei SHARE=NO vereinbart wird.

– Wird für eine Banddatei Kennwortschutz vereinbart, übertragen die Kennsatzverarbeitungsroutinen bei der Dateierstellung die Kennwörter vom Katalogeintrag in den HDR3-Kennsatz, ohne sie zu prüfen (umgekehrt werden bei Datei-Import Kennwörter vom HDR3-Kennsatz in den Katalogeintrag übertragen).
  Bei Dateiverarbeitung mit SECLEV=LOW wird die Prüfung der Kennwörter umgangen. Hat der Systemverwalter bei der Systemgenerierung Kennwortverschlüsselung vereinbart, wird bei Dateieröffnung im HDR3-Kennsatz das Verschlüsselungskennzeichen auf „1“ gesetzt.

– Soll eine Datei (FILE=...) umbenannt werden (NEWNAME=...), so darf der neue Name nur aus dem alten Namen bestehen, der mit einer in Klammern eingeschlossenen Versionsbezeichnung versehen ist. Die Versionsbezeichnung muss von einer evtl. ursprünglich vorhandenen Versionsbezeichnung verschieden sein.
  Diese Einschränkung ergibt sich aus der Bandkennsatz-Verarbeitung: Einzelne Blöcke einer Banddatei können aus Hardware-Gründen nicht überschrieben werden, und der Dateiname im Kennsatz wird bei der Datei-Eröffnung mit dem Dateinamen im Katalogeintrag verglichen.

– Zugriffsarten (ACCESS) für Banddateien:

  – Bei ACCESS=*WRITE sind alle OPEN-Modi zulässig.
  – Bei ACCESS=*READ sind nur die OPEN-Modi INPUT und REVERSE zulässig.
  – Entsprechend den Angaben im ACCESS-Operanden wird die Zugriffsart im HDR1-Kennsatz folgendermaßen eingetragen:
    ACCESS=*READ → Zugriffsart 1
    ACCESS=*WRITE → Zugriffsart 2
  – Der ACCESS-Operand wird hauptsächlich dazu verwendet, eine Datei gegen Zerstörung zu sichern (ACCESS=*READ). Nur der Eigentümer der Datei kann die Prüfung der Zugriffsart umgehen, indem er den Operanden SECLEV=LOW im FCB-Makro verwendet.

– Mehrbenutzbarkeit (USER-ACCESS/SHARE) für Banddateien:

  – Entsprechend den Angaben im SHARE-Operanden wird die Zugriffsart im HDR1-Kennsatz folgendermaßen eingetragen:
    SHARE=*NO (USER-ACCESS=*OWNER-ONLY) → Zugriffsvermerk 1
    SHARE=*YES (USER-ACCESS=*ALL-USERS) → Zugriffsvermerk 2
  – Es wird standardmäßig USER-ACCESS=*ALL-USERS angenommen, wenn der Katalogeintrag mit dem FILE-Aufruf erstellt wurde.

– DESTROY-Operand:
  Wurde DESTROY=*YES angegeben, wird nach dem Schließen (CLOSE) der Datei der Rest des Bandes gelöscht.

Hinweise zur Verwendung von Wildcard-Dateinamen

– Wenn ein Wildcard-Dateiname (Auswahl- und/oder Konstruktionsangabe) mit Stern beginnen soll und keine weiteren Wildcard-Symbole enthält, muss der führende Stern doppelt angegeben werden. Andernfalls wird der Auftrag abgewiesen.
  Beispiele: gültig sind '**A' und '*A/Z', ungültig sind '*ABC' und 'P(A)'.

– Wenn ein Wildcard-Dateiname (Auswahl und/oder Konstruktionsangabe) mit zwei Sternen beginnt, werden diese bezüglich der Konstruktion wie ein Stern gehandelt.
  Beispiel: Ein CATAL-Aufruf mit den Parametern FILE='A.*.* und NEWNAME='**.OLD.*' benennt eine existierende Datei 'A.TEST.1' um in 'TEST.OLD.1'.

– Der Inhalt der Class2-Option TEMPFILE stellt hier (im Gegensatz zu FSTAT) keinen teilqualifizierten Dateinamen dar. Stattdessen kann bzw. muss '.*' verwendet werden.

Hinweis zu SM-Pubsets

Folgende Angaben werden ignoriert für Dateien/Generationen/FGGs auf Volume-Sets mit permanenter Datenhaltung, wenn die Dateikennung am betroffenen SM-Pubset eine Default-Storage-Klasse hat und physikalische Allokierung verboten ist:
AVAIL, DISKWR, IOPERF, IOUSAGE, S0MIGR=*ALLOWED, STOCLAS=*NONE.

Funktionsübersicht

Funktion	FGG (Index)		Generation			permanente Datei			temp. Datei		Operand
	PUB	PRV	PUB	PRV	TAP	PUB	PRV	TAP	PUB	TAP	
Katalogeintrag identifizieren	x	x	x	x	x	x	x	x	x	x	pfadname
Datei/FGG umbenennen	x	x				x	x	x	x	x	$pfadname_2$
Katalogeintrag – erstellen – ändern – importieren	 x x	 x x x	 x x			 x x	 x	 x	 x x	 x	STATE =NEW =UPDATE =FOREIGN
Schreib- oder nur Lesezugriff erlauben	x	x				x	x	x			ACCESS
Zugriffskontrolle BASIC-ACL	x					x					BASACL
	x x x					x x x x					OWNERAR READ WRITE EXEC
	x x x					x x x x					GROUPAR READ WRITE EXEC
	x x x					x x x x					OTHERAR READ WRITE EXEC
Zugriffskontrolle GUARDS	x x x					x x x x					GUARDS READ WRITE EXEC
Übernahme von Schutzattributen	x	x				x	x	x	x	x	PROTECT
Mehrbenutzbarkeit	x	x				x	x	x			SHARE
Kennwortschutz gegen – Schreiben – Lesen – Ausführen	 x x	 x x				 x x x	 x x x	 x x x			 WRPASS RDPASS EXPASS
Schutzfrist	x	x				x	x				EXDATE (RETPD)
Freigabe zum Löschen	x	x				x	x				DELDATE

Funktion	FGG (Index)		Generation			permanente Datei			temp. Datei		Operand
	PUB	PRV	PUB	PRV	TAP	PUB	PRV	TAP	PUB	TAP	
automatische Datenzerstörung	x	x				x	x	x	x	x	DESTROY
Zugriffsüberwachung	x	x				x	x	x			AUDIT
Häufigkeit der ARCHIVE-Sicherung	x	x				x	x				BACKUP
Umfang der ARCHIVE-Sicherung	x	x				x	x				LARGE
HSMS-Migration zulassen/nicht zulassen	x	x				x	x	x			MIGRATE
SM-Pubset-Migration zulassen/nicht zulassen			x			x			x		S0MIGR
Verfügbarkeit			x			x					AVAIL
HSMS-Management-Klasse	x					x					MANCLAS
Storage-Klasse			x			x					STOCLAS
Zeichensatz zuweisen/ Zuweisung löschen	x					x		x	x	x	CCS
Performance-Eigenschaften festlegen			x			x			x		IOPERF
Bezugs-I/O-Operation für Performance			x			x			x		IOUSAGE
Eignung zur Bearbeitung im Cache (DAB)			x			x					DISKWR
benutzereigene Meta-Information	x		x			x			x		USRINFO
Systemverwaltungs-Meta-Information	x		x			x			x		ADMINFO
Dateigenerationsgruppe definieren	x	x									GEN
älteste Generation	x	x									FIRST
jüngste Generation	x	x									LAST
Bezugsgeneration	x	x									BASE
Überlaufbehandlung	x	x									DISP
Datenträger festlegen											
– Volume		x									VOLUME
– Device		x									DEVICE
Arbeitsgruppe	x										WORK

Legende

PUB : Public-Volume (gemeinschaftlicher Datenträger)

PRV : Private-Volume (Privatplatte)

TAP : Band

x: Das Attribut kann mit CATAL gesetzt bzw. modifiziert werden

Format

Operation	Operanden
CATAL	VERSION = 1 / 2 / 3 ,MF = C / D / E / L / M ,PARAM = <name 1..8> ,PREFIX = I/ <pre> ,MACID = DK/ <macid> ,ACCESS = *WRITE / *READ / *UNCHANGED ,*ACLPROT* = *NO / *YES ,*ADMINFO* = *NONE / <c-string 1..8> / (<reg: A(char:8)>) / <var: char:8> ,AUDIT = *NONE / *FAILURE / *SUCCESS / *ALL ,AVAIL = *STD / *HIGH ,BACKUP = *A / *B / *C / *D / *E ,BASACL = *NONE / *STD / *UNCHANGED ,BASE = <integer -99..9999> / (<reg: int:2>) / <var: int:2> ,CCS = *NONE / *STD / <c-string 1..8> / (<reg: A(char:8)>) / <var: char:8> ,CHECK = *NO / *MULTIPLE / *ERROR / *SINGLE / *CATALOG / *USERID ,DELDATE = *NONE / *UNCHANGED / <c-string 1..10> / (<reg: A(char:10)>) / <var: char:10> / [(<c-string 8..8> / (<reg: A(char:8)>) / <var: char:8>)] ,DESTROY = *NO / *YES / *UNCHANGED ,DEVICE = <c-string: device> / (<reg: A(char:8)>) / <var: char:8> ,DISKWR = *IMMEDIATE / *BY-CLOSE ,DISP = *CYCLE / *REUSE / *DELETE / *KEEP

(Teil 1 von 4)

Operation	Operanden
	,EXDATE = *UNCHANGED / <c-string 1..10> / (<reg: A(char:10)>) / <var: char:10> [(<c-string 8..8> / (<reg: A(char:8)>) / <var: char:8>)] ,EXPASS = *NONE / *UNCHANGED / <c-string 1..4> / <x-string 1..8>/ <integer -2147483648..2147483647> (<reg: A(char:4)>) / <var: char:4> ,FILE = <c-string 1..80: filename 1..54 with-wild(80)> / (<reg: A(char:80)>) / <var: char:80> ,FIRST = <integer 1..9999> / (<reg: int:2>) / <var: int:2> ,GEN = <integer 0..255> / (<reg: int:2>) / <var: int:2> ,GROUPAR = *NO-ACCESS / ([READ = *NO / READ = *YES / R = *N / R = *Y] [,WRITE = *NO / WRITE = *YES / W = *N / W = *Y] [,EXEC = *NO / EXEC = *YES / X= *N / X = *Y]) ,GUARDS = *NONE / ([READ = *NONE / <c-string: filename 1..18 without cat-gen-vers>/ <var: char:18> / (<reg: A(char:18)>)] [,WRITE = *NONE / <c-string: filename 1..18 without cat-gen-vers>/ <var: char:18> / (<reg: A(char:18)>)] [,EXEC = *NONE / <c-string: filename 1..18 without cat-gen-vers>/ <var: char:18> / (<reg: A(char:18)>)]) / *UNCHANGED ,IOPERF = *STD / *HIGH / *VERY-HIGH / *USER-MAX ,IOUSAGE = *READ-WRITE / *WRITE / *READ ,LARGE = *NO / *YES ,LAST = <integer 1..9999> / (<reg: int:2>) / <var: int:2> ,LIST = *NO / *SYSOUT / *ERRORS-TO-SYSOUT ,MANCLAS = *NONE / <c-string: struct-name 1..8> / (<reg: A(char:8)>) / <var: char:8>

(Teil 2 von 4)

Operation	Operanden
	,MIGRATE = *ALLOWED / *INHIBITED / *FORBIDDEN ,NEWNAME = c-string 1..80: filename 1..54 with-constr-wild(80)> / (<reg: A(char:80)>) / <var: char:80> ,OPNBACK = *NO / *YES ,OTHERAR = *NO-ACCESS / ([READ = *NO / READ = *YES / R = *N / R = *Y] [,WRITE = *NO / WRITE = *YES / W = *N / W = *Y] [,EXEC = *NO / EXEC = *YES / X= *N / X = *Y]) ,OWNERAR = *NO-ACCESS / ([READ = *NO / READ = *YES / R = *N / R = *Y] [,WRITE = *NO / WRITE = *YES / W = *N / W = *Y] [,EXEC = *NO / EXEC = *YES / X= *N / X = *Y]) ,PROTECT = *STD / *BY_DEF_PROT_OR_STD / (*FROM_FILE,<c-string: filename 1..54>) / (*FROM_FILE,(<reg: A(char:54)>)) / (*FROM_FILE,<var: char:54>) ,RDPASS = *NONE / *UNCHANGED / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647> (<reg: A(char:4)>) / <var: char:4> ,RELSPAC = *ALLOWED / *IGNORED / *UNCHANGED ,RETPD = <integer 0..32767> / (<reg: int:2>) / <var: int:2> ,SOMIGR = *ALLOWED / *FORBIDDEN ,SHARE = *NO / *YES / *SPECIAL / *UNCHANGED ,STATE = *NEW / *UPDATE / *FOREIGN ,STOCLAS = *STD / *NONE / *UPDATE / <c-string: struct-name 1..8>/ (<reg: A(char:8)>) / <var: char:8> ,TIMBASE = *UTC / *LTI ,USRINFO = *NONE / <c-string 1..8> / (<reg: A(char:8)>) / <var: char:8> ,VOLUME = <c-string: vsn 1..6> / (<reg: A(char:6)>) / <var: char:6>

(Teil 3 von 4)

Operation	Operanden
	,WORKGRP = *YES ,WRPASS = *NONE / *UNCHANGED / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647> (<reg: A(char:4)>) / <var: char:4>

(Teil 4 von 4)

Operandenbeschreibung

ACCESS
Mit dem ACCESS-Operanden kann eine Datei gegen Überschreiben gesichert werden; er gibt an, ob auf die Datei oder Dateigenerationen nur lesend oder auch schreibend zugegriffen werden darf. Dieses Schutzattribut ist nur relevant, wenn kein BASIC-ACL- oder GUARDS-Schutz aktiviert ist.

Banddateien:
Das DVS übernimmt beim ersten Eröffnen der Datei das ACCESS-Kennzeichen in den HDR3-Kennsatz. Bei späteren Dateizugriffen kann der Dateieigentümer die Überprüfung der Zugriffsart durch SECLEV=LOW (siehe Seite 441 bzw. Seite 495) umgehen.

Voreinstellung – nur bei STATE=*NEW: ACCESS=*WRITE

= *WRITE
Für die Datei oder die Dateigenerationen sind alle Zugriffsarten zugelassen.
Banddateien, HDR3-Kennsatz: Zugriffsart = 0

= *READ
Auf die Datei oder die Dateigenerationen darf nur lesend zugegriffen werden, es sind nur die OPEN-Modi INPUT und REVERSE zulässig.
temporäre Dateien: Schreibzugriff kann nicht unterbunden werden, ACCESS=READ wird abgewiesen.
Banddateien, HDR3-Kennsatz: Zugriffsart = 1

= *UNCHANGED

Nur relevant im Zusammenhang mit der Angabe von PROTECT:
Bei gleichzeitiger Angabe von STATE=*UPDATE bleibt der Wert für ACCESS unverändert, bei gleichzeitiger Angabe von STATE=*NEW wird der Wert ACCESS=*WRITE eingetragen.

Der Wert *UNCHANGED unterdrückt insbesondere bei
– PROTECT=*FROM_FILE
  die Übernahme des entsprechenden Wertes aus der Referenzdatei
– PROTECT=*BY_DEF_PROT_OR_STD und bei STATE=*NEW ohne Angabe zu PROTECT
  die Übernahme des entsprechenden von der Funktion „Default-Protection“ gelieferten Wertes
– gleichzeitiger Angabe von PROTECT=*STD und STATE=*UPDATE
  das Zurücksetzen des Wertes im Katalogeintrag auf den Wert ACCESS=*WRITE

Bei STATE=*UPDATE gilt: Falls PROTECT nicht angegeben ist, ist die Angabe von *UNCHANGED gleichbedeutend mit fehlender Angabe.

Bei STATE=*NEW gilt: Die Angabe *UNCHANGED ist gleichbedeutend mit der Angabe ACCESS=*WRITE (unabhängig von der Angabe bei PROTECT).

ACLPROT
Nur für Systemverwalter zulässig (Aufruf unter der Kennung TSOS):
setzt den ACL-Indikator oder setzt ihn zurück und schaltet so die Zugriffskontrolle über ACL ein und aus.

Seit SECOS V4.0 wird die Zugriffskontrolle über ACL nicht mehr unterstützt. Stattdessen ist der Zugriffsschutz mit GUARDS zu verwenden.

Dieser Operand kann zu Inkonsistenzen im System führen und darf deshalb nur verwendet werden, um den Zugriff auf eine ACL-geschützte Datei zu ermöglichen.
Der Operand ist nicht erlaubt für temporäre Dateien, Banddateien und Dateien auf privaten Datenträgern.

= *NO

Der ACL-Indikator wird zurückgesetzt, die Zugriffskontrolle über ACL damit ausgeschaltet.

= *YES

Sollte nicht mehr verwendet werden, da ACL-geschützte Dateien nicht zugreifbar sind:
Der ACL-Indikator wird gesetzt. Ein ggf. vereinbarter GUARDS-Schutz wird gelöscht. Falls die Funktion „Default-Protection“ einen Wert für GUARDS liefert, wird dieser ignoriert.

ADMINFO
Nur für Systemverwalter zulässig (Aufruf unter der Kennung TSOS):
Trägt eine Systemverwaltungs-Metainformation in den Katalogeintrag der Datei ein. Der Eintrag kann maximal 8 Byte beliebigen Inhalts aufnehmen; die Bedeutung legt der Systemverwalter fest.
Für Dateien auf privaten Datenträgern wird der Operand ignoriert.

= *NONE
Der Eintrag wird gelöscht.

= <c-string 1..8>
Die angegebenen Zeichen werden eingetragen.

= (<reg: A(char:8)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 8 Byte, in dem die einzutragende Metainformation abgelegt ist.

= <var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 8 Byte, in dem die einzutragende Metainformation abgelegt ist.

AUDIT
Nur für Benutzerkennungen mit dem Recht AUDIT=YES:
bestimmt, ob DVS-Zugriffe auf Dateien/Dateigenerationen durch System-Exit-Routinen überwacht werden sollen. Die Überwachung bezieht sich auf die Operationen CATAL, FILE, OPEN und ERASE.
Hat der Anwender nicht die Berechtigung AUDIT=YES, wird ein CATAL-Makroaufruf mit AUDIT-Angabe abgewiesen.

Voreinstellung – nur bei STATE=*NEW: AUDIT=*NONE

= *NONE
Keine Überwachung

= *ALL
Alle DVS-Operationen für die Datei/Generationen werden überwacht.

= *SUCC
Alle erfolgreichen DVS-Operationen für die Datei/Generationen werden überwacht.

= *FAIL
Alle erfolglosen DVS-Operationen für die Datei/Generationen werden überwacht.

AVAIL

*Nur relevant bei STATE=*UPDATE für Dateien auf öffentlichen Datenträgern oder auf einem Net-Storage-Volume:*

Die Anforderungen bezüglich der Ausfallsicherheit der Datei werden modifiziert. Dateien, die eine erhöhte Ausfallsicherheit haben sollen, werden auf einem entsprechenden Volume-Set (z.B. DRV – Dual Recording by Volume) allokiert.
Im SM-Pubset wird durch die Angabe des Operanden eine eventuell im Katalogeintrag der Datei eingetragene Storage-Klasse entfernt.

Der Operand darf nicht gleichzeitig mit dem Operanden STOCLAS≠*NONE angegeben werden.

Der Operand wird ignoriert für Dateien und Generationen auf Volume-Sets mit permanenter Datenhaltung, wenn die Dateikennung auf dem entsprechenden SM-Pubset eine Default-Storage-Klasse besitzt und physikalische Allokierung verboten ist.

= *STD

Es werden keine besonderen Anforderungen bezüglich der Ausfallsicherheit gestellt.

= *HIGH

Die Datei soll eine erhöhte Ausfallsicherheit haben. Es wird sichergestellt, dass die Datei auf einem entsprechenden Volume-Set allokiert ist. Bietet der gegenwärtige Ablageort nicht die Anforderungen einer erhöhten Ausfallsicherheit, wird der Auftrag für SF-Pubsets abgewiesen. In einem SM-Pubset wird der Auftrag nur dann abgewiesen, wenn kein geeigneter Volume-Set verfügbar ist oder die zulässigen Kontingente der Benutzerkennung überschritten werden. Ansonsten wird der Speicherplatz auf einen geeigneten Volume-Set umallokiert.

Ist die Datei auf einem privaten Datenträger abgelegt, ist sie eine Arbeitsdatei (WORK-FILE), oder ist sie auf eine Hintergrundebene migriert, dann wird der Auftrag mit Returncode abgewiesen. Bei temporären Dateien wird der Auftrag ebenfalls abgewiesen. Dies gilt auch, wenn eine temporäre Datei in eine permanente Datei umbenannt wird. In diesem Fall muss erst das Umbenennen erfolgen, und erst dann kann mit einem weiteren CATAL-Aufruf erhöhte Ausfallsicherheit vereinbart werden.

Hinweis für Dateien in SM-Pubsets:

Wenn der derzeitige Ablageort (Volume-Set) nicht die Anforderung einer hohen Verfügbarkeit bietet, aber ein geeigneter Volume-Set im SM-Pubset vorhanden ist, erfolgt automatisch ein Umallokieren der Daten bzw. des Speicherplatzes. Während des Umallokierens ist die Datei gesperrt (geöffnet), d.h. alle Zugriffe auf die Datei bzw. ihren Katalogeintrag werden abgewiesen, also insbesondere nicht in einen Wartezustand versetzt.

BACKUP
Nur für Dateien/FGG auf Platten:
BACKUP bezieht sich auf die automatische Dateisicherung mit dem Sicherungssystem ARCHIVE bzw. HSMS und legt fest, bei welchen Sicherungsläufen die Dateien oder die Generationen der FGG gesichert werden sollen.

Voreinstellung: für permanente Dateien: gemäß Class2-Option BACKUP
für temporäre Dateien: BACKUP=*E

= *A
Die Dateien/Generationen werden bei jedem Sicherungslauf gesichert.

= *B
Die Dateien/Generationen werden gesichert, wenn ein Sicherungslauf für Dateien mit BACKUP=*B/*C/*D stattfindet.

= *C
Die Dateien/Generationen werden gesichert bei Sicherungsläufen für BACKUP=*C/*D.

= *D
Dateien/Generationen werden nur gesichert bei Sicherungsläufen für BACKUP=*D.

= *E
keine automatische Sicherung mit ARCHIVE

BASACL
aktiviert die Zugriffskontrolle durch BASIC-ACL mit Standardzugriffsrechten oder deaktiviert die Zugriffskontrolle durch BASIC-ACL.
Der Schutz ist nur wirksam, wenn kein GUARDS-Schutz aktiviert ist.

Der Operand darf nicht zusammen mit den Operanden OWNERAR, GROUPAR oder OTHERAR angegeben werden.

Voreinstellung – nur bei STATE=*NEW: BASACL=*NONE

= *NONE
deaktiviert die Zugriffskontrolle durch BASIC-ACL.

= *STD
trägt die im Folgenden beschriebenen Standardzugriffsrechte in die Basic-ACL ein:

– Bei Angabe von STATE=*NEW bzw. fehlender STATE-Angabe werden folgende Zugriffsrechte eingetragen:

```
OWNER   GROUP   OTHERS
R W X   - - -   - - -
```

Dies entspricht der Angabe der Operanden OWNERAR=(READ=*YES,WRITE=*YES,EXEC=*YES), GROUPAR=*NO-ACCESS, OTHERAR=*NO-ACCESS.

– Bei Angabe von STATE=*UPDATE (d.h. bei einer bereits katalogisierten Datei) werden die gültigen Werte für SHARE und ACCESS in BASIC-ACL-Werte umgesetzt, falls die Zugriffskontrolle über Basic ACL bisher nicht aktiviert war. Für die Umsetzung gilt folgende Tabelle:

SHARE	ACCESS	OWNER			GROUP			OTHERS		
		R	W	X	R	W	X	R	W	X
NO	READ	R	-	X	-	-	-	-	-	-
NO	WRITE	R	W	X	-	-	-	-	-	-
YES / SPECIAL	READ	R	-	X	R	-	X	R	-	X
YES / SPECIAL	WRITE	R	W	X	R	W	X	R	W	X

Hinweise

Werden zusammen mit BASACL=*STD auch die Operanden SHARE, ACCESS und/oder PROTECT angegeben, dann erfolgt die Abbildung unter Berücksichtigung dieser Angaben.
Bei Dateigenerationsgruppen wird für EXEC nichts (kein Zugriff) eingetragen.

= *UNCHANGED

Nur relevant im Zusammenhang mit der Angabe von PROTECT:
Bei gleichzeitiger Angabe von STATE=*UPDATE bleibt der Wert für BASIC-ACL unverändert, bei gleichzeitiger Angabe von STATE=*NEW wird kein BASIC-ACL eingetragen.

Der Wert *UNCHANGED unterdrückt insbesondere bei
– PROTECT=*FROM_FILE
die Übernahme des entsprechenden Wertes aus der Referenzdatei
– PROTECT=*BY_DEF_PROT_OR_STD und bei STATE=*NEW ohne Angabe zu PROTECT
die Übernahme des entsprechenden von der Funktion „Default-Protection“ gelieferten Wertes
– gleichzeitiger Angabe von PROTECT=*STD und STATE=*UPDATE
das Zurücksetzen des Wertes im Katalogeintrag (kein BASIC-ACL)

Bei STATE=*UPDATE gilt: Falls PROTECT nicht angegeben ist, ist die Angabe von *UNCHANGED gleichbedeutend mit fehlender Angabe.

Bei STATE=*NEW gilt: Die Angabe *UNCHANGED ist gleichbedeutend mit kein BASIC-ACL im Katalogeintrag (unabhängig von der Angabe bei PROTECT).

BASE

Nur für Dateigenerationsgruppen:

Definiert einen Bezugspunkt (eine Basisgeneration), auf die sich alle relativen Generationsnummern beziehen und ermöglicht außerdem die Rekonstruktion eines Index für Dateigenerationen auf privaten Datenträgern.

Wird beim Erstellen einer FGG (Gruppenindex) der BASE-Operand nicht angegeben, dann erhält er den Wert des FIRST-Operanden, bzw. wenn dieser auch nicht angegeben wird, den Wert 0.

Anwendung zur Index-Rekonstruktion:

Sollen von einer Privatplatte Generationen einer FGG importiert werden, deren Gruppeneintrag weder im Katalog noch im F1-Etikett der Platte existiert, muss zunächst der Gruppenindex rekonstruiert werden. Dazu muss ein CATAL-Aufruf mit den Operanden STATE=*NEW, GEN=<zahl>, FIRST=<zahl> und mindestens einem der Operanden BASE oder LAST durchgeführt werden. Wird nur der BASE-Operand angegeben, bestimmt er gleichzeitig die jüngste Dateigeneration, d.h. den Wert für LAST.

= <integer -99..9999>

Die neue Basisgeneration kann in Abhängigkeit vom angegebenen Wert für „zahl“ in absoluter oder relativer Form vereinbart werden.

absolute Form: Wertebereich: 1 ≤ zahl ≤ 9999.

– STATE=*UPDATE: „zahl“ wird der neue Basiswert, der eine laut Indexeintrag existierende Generation bezeichnen muss.
– STATE=*NEW: „zahl“ wird als Basiswert in den Katalog aufgenommen.
  Wird BASE im Zusammenhang mit FIRST und/oder LAST angegeben, dann muss die Bedingung FIRST ≤ zahl ≤ LAST erfüllt sein.

relative Form:

– *Diese Angabe ist nur mit STATE=*UPDATE möglich:*
  Wertebereich: -99 ≤ zahl ≤ 0.
  Bestimmt die Basisgeneration relativ zur jüngsten, laut Index katalogisierten Generation (Katalogfeld `LAST-GEN`).
  Der neue Basiswert muss eine laut Index existierende Dateigeneration bezeichnen, d.h. er muss ≥ dem Wert im Ausgabefeld `FIRST-GEN` sein.

= (<reg: int:2>)

Nur mit MF=M möglich:

Das angegebene Register enthält im unteren Halbwort den Basiswert (Wertebereich und Bedeutung siehe oben).

= <var: int:2>

Nur mit MF=M möglich:

Symbolische Adresse eines Halbwortes, in dem der Basiswert abgelegt ist (Wertebereich und Bedeutung siehe oben).

CCS
Nur für Dateien/FGG auf gemeinschaftlichen Datenträgern, für Dateien auf Net-Storage und für Banddateien:
Gültiger Zeichensatz der Datei.
In der Codiertabelle (Coded Character Set) ist festgelegt, wie die Zeichen eines nationalen Zeichensatzes binär abzuspeichern sind. Der festgelegte Zeichensatz beeinflusst z.B. die Bildschirmdarstellung der Zeichen, Sortierreihenfolge usw. (siehe Handbuch „XHCS“ [22]).

Für Dateien auf Privatplatte wird der Operand ignoriert, also auch kein Returncode geliefert.

Voreinstellung – nur bei STATE=*NEW: CCS=*NONE

= *NONE
Für die Datei soll kein Zeichensatz festgelegt werden.

= *STD
Der Zeichensatz wird aus dem Benutzerkatalogeintrag des Dateieigentümers übernommen, falls dort ein Zeichensatz ungleich `EDF03IRV` eingetragen ist. Anderfalls gilt *NONE.

= <c-string 1..8>
Name des Zeichensatzes, mit dem die Datei bearbeitet werden soll (z.B. `EDF03IRV` für die internationale Version von `EBCDIC.DF.03`).
Der angegebene String wird nicht überprüft, insbesondere nicht dahingehend, ob es sich um den Namen eines definierten Coded Character Sets handelt.

= (<reg: A(char:8)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 8 Byte, in dem der Name des Coded Character Sets abgelegt ist.

= <var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 8 Byte, in dem der Name des Coded Character Sets abgelegt ist.

CHECK
Legt fest, unter welchen Bedingungen ein Anwenderdialog gestartet werden soll.

Wird der Dialog gestartet, kann der Benutzer entscheiden, ob die Verarbeitung für die angezeigten Dateien ausgeführt werden soll oder nicht. Er kann sich außerdem einen Hilfetext zu den Antwortmöglichkeiten ausgeben lassen sowie bei Fortsetzung der Verarbeitung einen neuen Wert für LIST und/oder CHECK festlegen.

Im Stapelbetrieb gilt immer der Wert *NO.

Voreinstellung: CHECK=*NO

= *NO
Alle ausgewählten Dateien werden ohne Kontrolldialog, d.h. ohne Eingriffsmöglichkeit des Benutzers verarbeitet.

= *MULTIPLE
Ein Kontrolldialog wird nur gestartet, wenn mehrere Dateien ausgewählt sind.
Sind Wildcards in der Katalog und/oder Benutzerkennung enthalten, dann wird für jeden Katalog und/oder jede Benutzerkennung ein Kontrolldialog durchgeführt.
Implizit gilt auch CHECK=*ERROR.

= *ERROR
Ein Fehler-Kontrolldialog wird gestartet, wenn bei der Verarbeitung eines ausgewählten Dateinamens ein Fehler auftritt. Ein Dateimenge-Kontrolldialog wird gestartet, wenn die Auswahlangabe mehr Dateien selektiert, als Speicherplatz für deren Verarbeitung verfügbar ist. Bei allen CHECK-Angaben ≠ *NO gilt implizit auch immer CHECK=*ERROR!

= *SINGLE
Für jeden ausgewählten Dateinamen wird ein Kontrolldialog durchgeführt. Implizit gilt auch CHECK=*ERROR.

= *CATALOG
Der Anwender muss in einem Kontrolldialog für jeden Katalog entscheiden, ob die darauf ausgewählten Dateien verarbeitet werden sollen.
Implizit gilt auch CHECK=*ERROR.

= *USERID
nur für Systemverwalter:
Der Systemverwalter muss in einem Kontrolldialog für jede Benutzerkennung auf jedem Katalog entscheiden, ob die ausgewählten Dateien verarbeitet werden sollen.
Implizit gilt auch CHECK=*ERROR.

DELDATE
Nur für Dateien auf gemeinschaftlichem Datenträger und für Dateien auf Net-Storage:
Legt den Zeitpunkt fest, ab dem die Datei ohne Berücksichtigung der Schutzattribute ACCESS, BASACL, EXDATE, GUARDS, RDPASS, WRPASS und EXPASS gelöscht oder ihr Speicherplatz freigegeben werden darf.

Eine absolute Datumsangabe wird abhängig von der Angabe des Operanden TIMBASE auf Basis der lokalen Zeit LTI (local time) oder der Weltzeit UTC (universal time coordinate) interpretiert, eine relative Angabe stets auf Basis der lokalen Zeit.

= *NONE
Die Datei soll nicht ohne Berücksichtigung der Schutzattribute gelöscht werden können (entspricht DELDATE='+0').

= *UNCHANGED
Bei gleichzeitiger Angabe von STATE=*UPDATE bleibt der Wert für DELDATE unverändert, bei gleichzeitiger Angabe von STATE=*NEW wird der Wert DELDATE=*NONE eingetragen.

Bei STATE=*UPDATE gilt: Falls PROTECT=*BY_DEF_PROT_OR_STD angegeben ist, unterdrückt der Wert *UNCHANGED die Übernahme des entsprechenden von der Funktion „Default-Protection“ gelieferten Wertes; andernfalls ist die Angabe *UNCHANGED gleichbedeutend mit fehlender Angabe.

Bei STATE=*NEW gilt: Die Angabe *UNCHANGED ist gleichbedeutend mit der Angabe *NONE (unabhängig von der Angabe bei PROTECT).

= <c-string 1..10> / (<reg: A(char:10)>) / <var: char:10>
[(<c-string 8..8> / (<reg: A(char:8)>) / <var: char:8>)]
Bestimmt den Zeitpunkt, ab dem die Datei ohne Berücksichtigung der Schutzattribute gelöscht werden darf. Dabei stehen die Variablen „...10“ für Datumsangaben und „...8“ für Zeitangaben.

Folgende Formate sind erlaubt:
– '+<integer 0..99999>'[('<time 8..8>')]
– '<date 8..10>'[('<time 8..8>')]
– 'yymmdd'[('<time 8..8>')]

Bei der relativen Angabe ist zur Unterscheidung von den absoluten Datumsangaben unbedingt das '+' anzugeben.

Datumsangaben, die nicht die volle Länge von 10 Zeichen ausfüllen, müssen mit Leerzeichen abgeschlossen sein.

Zweistellige Jahresangaben von 00 bis 59 werden mit 20 ergänzt, von 60 bis 99 mit 19.

Beispiele

```
          ...
          LA    7,DELTIME
BEISP1    CATAL MF=M,VERSION=3,...,DELDATE='+1'((7))
          LA    6,DELDATE
BEISP2    CATAL MF=M,VERSION=3,...,DELDATE=(6)(DELTIME)
BEISP3    CATAL MF=M,VERSION=3,...,DELDATE=DELDATE('00:00:00')
          ...
DELDATE   DC    CL10'2011-11-11'    11.11.2011
DELTIME   DC    CL8'11:11'          11:11:00
```

DESTROY
Zur Erhöhung des Datenschutzes kann der Benutzer im Katalogeintrag festlegen, dass nicht mehr benötigte Daten mit X'00' (binär null) überschrieben werden. Bei Plattendateien wirkt sich die DESTROY-Angabe bei Speicherplatzfreigabe und Löschen aus (FILE- und ERASE-Makro), bei Banddateien auf das Überschreiben von Restdaten bei EOF- und EOV-Verarbeitung (siehe Makro FILE, Operand DESTOC, Seite 478).

Voreinstellung – nur bei STATE=*NEW: DESTROY=*NO

= NO
Plattendateien:
Der Speicherplatz wird unverändert freigegeben, wenn nicht im ERASE-Aufruf der Operand DESTROY=*YES angegeben wurde.

Banddateien:
Auf dem Band folgende Restdaten werden nicht überschrieben, wenn im FILE-Aufruf für den aktuellen Verarbeitungslauf nicht DESTOC=YES vereinbart wird.

= YES
Plattendateien:
Der Speicherplatz wird bei Freigabe automatisch mit binär null (X'00') überschrieben.

Banddateien:
Restdaten auf Band werden gelöscht; mit FILE kann ebenfalls Löschen der Restdaten für den aktuellen Verarbeitungslauf eingestellt werden (Operand DESTOC).

= *UNCHANGED
Nur relevant im Zusammenhang mit der Angabe von PROTECT:
Bei gleichzeitiger Angabe von STATE=*UPDATE bleibt der Wert für DESTROY unverändert, bei gleichzeitiger Angabe von STATE=*NEW wird der Wert DESTROY=*NO eingetragen.

Der Wert *UNCHANGED unterdrückt insbesondere bei
– PROTECT=*FROM_FILE
  die Übernahme des entsprechenden Wertes aus der Referenzdatei
– PROTECT=*BY_DEF_PROT_OR_STD und bei STATE=*NEW ohne Angabe zu PROTECT
  die Übernahme des entsprechenden von der Funktion „Default-Protection“ gelieferten Wertes
– gleichzeitiger Angabe von PROTECT=*STD und STATE=*UPDATE
  das Zurücksetzen des Wertes im Katalogeintrag auf den Wert DESTROY=*NO

Bei STATE=*UPDATE gilt: Falls PROTECT nicht angegeben ist, ist die Angabe von *UNCHANGED gleichbedeutend mit fehlender Angabe.

Bei STATE=*NEW gilt: Die Angabe *UNCHANGED ist gleichbedeutend mit der Angabe DESTROY=*NO (unabhängig von der Angabe bei PROTECT).

DEVICE
Nur für Dateigenerationsgruppen auf Privatplatten in Zusammenhang mit dem Operanden VOLUME:
Gibt an, auf welchem Gerätetyp die Dateigenerationsgruppe gespeichert bzw. von welchem Gerätetyp sie importiert werden soll (vgl. FILE-Makro, DEVICE-Operand).

Ist MAREN verfügbar, muss DEVICE nicht angegeben werden.

= <c-string: device>
Gerätetyp; mögliche Angaben sind der Gerätetabelle (siehe Handbuch „Systeminstallation [16]) zu entnehmen.
Jede Angabe eines Plattengerätetyps wird wie die Angabe STDDISK behandelt.

= (<reg: A(char:8)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 8 Byte, in dem der Gerätetyp abgelegt ist.

= <var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 8 Byte, in dem der Gerätetyp abgelegt ist.

DISKWR
Nur für Dateien und Dateigenerationen auf gemeinschaftlichen Datenträgern sowie Dateien auf Net-Storage-Volumes:
Gibt an, zu welchem Zeitpunkt nach einer Schreiboperation Datenkonsistenz gefordert wird.
Bei der Bearbeitung über einen flüchtigen Schreib-Cache befinden sich die Daten der Datei erst nach der CLOSE-Verarbeitung in einem konsistenten Zustand. Systemfehler während der Bearbeitungsphase können zu Inkonsistenzen führen. Für Dateien, die wichtige Daten enthalten, sollte deshalb Datenkonsistenz nach jeder Schreiboperation gefordert werden.

Voreinstellung:
DISKWR=*IMMEDIATE für permanente Dateien
DISKWR=*BY-CLOSE für temporäre Dateien

Wird DISKWR beim Umkatalogisieren temporär → permanent bzw. permanent → temporär nicht angegeben, dann wird der Eintrag automatisch auf den jeweiligen Defaultwert/erlaubten Wert gesetzt.

Der Operand wird ignoriert (kein Returncode!), wenn:
– er für Dateien angegeben wird, die nicht auf einem gemeinschaftlichen Datenträger liegen bzw. angelegt werden sollen.
– die Dateikennung auf dem entsprechenden SM-Pubset eine Default-Storage-Klasse besitzt und physikalische Allokierung verboten ist.

Im SM-Pubset wird durch die Angabe des Operanden eine evtl. im Katalogeintrag der Datei eingetragene Storage-Klasse entfernt. Der Operand darf nicht gleichzeitig mit dem Operanden STOCLAS≠*NONE angegeben werden.

= *IMMEDIATE
Die Daten der Datei müssen sich direkt nach Beendigung einer Schreiboperation in konsistentem Zustand befinden, d.h. die Datei soll nicht über einen flüchtigen Schreib-Cache bearbeitet werden (Defaultwert für permanente Dateien).
Für temporäre Dateien wird diese Angabe ignoriert.

= *BY-CLOSE
Die Daten der Datei müssen sich erst nach der CLOSE-Verarbeitung in einem konsisten Zustand befinden (Defaultwert für temporäre Dateien).

DISP
Nur für Dateigenerationsgruppen:
Legt fest, ob beim Überschreiten der mit GEN festgelegten Höchstzahl gleichzeitig existierender Generationen die jeweils älteste Generation gelöscht und ihr Speicherplatz evtl. wiederverwendet werden soll. Bei Generationen auf Band wird nur der Katalogeintrag gelöscht.
Eine bestehende Schutzfrist wird beim Löschen der ältesten Generation ignoriert.

Voreinstellung – nur bei STATE=*NEW: DISP=*CYCLE

= *CYCLE
Die jeweils älteste Generation wird gelöscht, ihr Speicherplatz bzw. die von ihr belegten Bänder werden freigegeben. Im Gruppeneintrag werden die Felder LAST-GEN und FIRST-GEN (jüngste/älteste existente Generation) aktualisiert.

= *REUSE
Die Auswirkung von DISP=*REUSE ist abhängig vom Speichermedium:

Für FGG auf gemeinschaftlichen Platten:
Die älteste Generation wird gelöscht, ihr Speicherplatz an das System zurückgegeben, der Gruppeneintrag aktualisiert (siehe DISP=*CYCLE).

Für FGG auf Privatplatte:
Die neue Generation wird eingerichtet und die älteste Generation gelöscht. Der Datenträger der ältesten Generation wird für die Speicherung der neuen Generation verwendet. Erstreckte sich die gelöschte Generation über mehrere Platten, wird die neue Generation nur auf der ersten Platte katalogisiert. Der Gruppeneintrag wird entsprechend aktualisiert. Da die alte Generation erst gelöscht wird, wenn die neue Generation eingerichtet ist, kann Speicherplatzmangel auf der Platte dazu führen, dass die neue Generation nicht eingerichtet werden kann, obwohl DISP=*REUSE gilt.

für FGG auf Band:
Die älteste Generation wird aus dem Katalog gelöscht, die neue Generation wird auf den frei werdenden Bändern eingerichtet. Der Gruppeneintrag wird entsprechend aktualisiert. DISP=*REUSE ist nicht zulässig für Dateigenerationsgruppen auf MF/MV-Sets.

= *DELETE
Alle Generationen der FGG werden gelöscht, die neue Generation wird zur ältesten der neuen Serie. Der Gruppeneintrag wird entsprechend aktualisiert.

= *KEEP
Die „überzähligen“ ältesten Generationen werden nicht automatisch gelöscht, sondern erst dann, wenn der Anwender mit einem erneuten CATAL-Aufruf mit den Operanden FIRST und BASE eine neue „älteste“ und eine neue Basisgeneration bestimmt oder wenn er mit DISP= einen neuen Wert angibt. Im Gruppeneintrag wird beim Erstellen neuer Generationen jeweils nur das Feld LAST-GEN aktualisiert.

EXDATE
Legt die Schutzfrist (EXPIRATION-DATE) fest, während der die Datei nicht geändert oder gelöscht werden kann; d.h. sie kann nur gelesen werden („read only“).

Eine Schutzfrist kann nur für existente Dateien vereinbart werden, d.h. die Katalogfelder CRE-DATE und FILE-STRUC müssen einen Wert ≠ NONE anzeigen. Das heißt auch, die CATAL-Operanden EXDATE und STATE=*NEW bzw. STATE=*FOREIGN sind nicht kombinierbar (EXDATE wird ignoriert).

Eine absolute Datumsangabe wird abhängig von der Angabe des Operanden TIMBASE auf Basis der lokalen Zeit LTI (local time) oder der Weltzeit UTC (universal time coordinate) interpretiert, eine relative Angabe stets auf Basis der lokalen Zeit.

Im Zusammenhang mit TIMBASE=*LTI bzw. einer relativen Datumsangabe wird als Uhrzeit immer 00:00:00 angenommen. Die explizite Angabe einer Uhrzeit wird nur im Zusammenhang mit TIMBASE=*UTC berücksichtigt, wobei jedoch die Minuten und Sekunden stets auf null gesetzt werden.

Hinweis

Ein vor dem aktuellen Tagesdatum liegendes Ende der Schutzfrist wird nicht eingetragen, stattdessen das aktuelle Tagesdatum mit der lokalen Zeit 0.00 Uhr.

Eine gleichzeitige Verwendung der Operanden EXDATE und RETPD ist nicht möglich.

= <c-string 1..10> / (<reg: A(char:10)>) / <var: char:10>
[(<c-string 8..8> / (<reg: A(char:8)>) / <var: char:8>)]
Bestimmt den Zeitpunkt, ab dem die Datei wieder verändert werden kann. Dabei stehen die Variablen „...10“ für Datumsangaben und „...8“ für Zeitangaben.

Folgende Formate sind erlaubt:
– '+<integer 0..99999>'
– '<date 8..10>'[('<time 8..8>')]
– 'yymmdd'[('<time 8..8>')]

Bei der relativen Angabe ist zur Unterscheidung von den absoluten Datumsangaben unbedingt das „+“ anzugeben.

Datumsangaben, die nicht die volle Länge von 10 Zeichen ausfüllen, müssen mit Leerzeichen abgeschlossen sein.

Zweistellige Jahresangaben von 00 bis 59 werden mit 20 ergänzt, von 60 bis 99 mit 19.

Beispiele

```
          ...
          LA    6,EXPDATE
          LA    7,EXPTIME
BEISP1    CATAL MF=M,VERSION=3,...,EXDATE=(6)((7))
BEISP2    CATAL MF=M,VERSION=3,...,EXDATE=(6)('23:00:00')
BEISP3    CATAL MF=M,VERSION=3,...,EXDATE=EXPDATE(EXPTIME)
          ...
EXPDATE   DC    CL10'111231'        31.12.2011
EXPTIME   DC    CL8'00:00:00'       00:00:00
```

= *UNCHANGED

Nur relevant im Zusammenhang mit der Angabe von PROTECT:
Bei gleichzeitiger Angabe von STATE=*UPDATE bleibt der Wert für EXDATE unverändert.

Der Wert *UNCHANGED unterdrückt für permanente Dateien mit Erstellungsdatum und für Dateigenerationsgruppen mit bereits katalogisiertem Gruppeneintrag insbesondere bei
– PROTECT=*FROM_FILE
  die Übernahme des entsprechenden Wertes aus der Referenzdatei
– PROTECT=*BY_DEF_PROT_OR_STD
  die Übernahme des entsprechenden von der Funktion „Default-Protection“ gelieferten Wertes
– PROTECT=*STD
  das Zurücksetzen des Wertes im Katalogeintrag auf das aktuelle Tagesdatum

Bei STATE=*UPDATE gilt: Falls PROTECT nicht angegeben ist, ist die Angabe von *UNCHANGED gleichbedeutend mit fehlender Angabe.

EXPASS
Nur für Dateien, nicht für FGG/Dateigenerationen:
Der Benutzer kann mit EXPASS ein sog. „Ausführungskennwort“ definieren oder löschen. Der „Ausführungs“-Schutz bezieht sich auf den Aufruf eines Programms oder einer Prozedurdatei mit den Kommandos START-PROGRAM, LOAD-PROGRAM, CALL-PROCEDURE, INCLUDE-PROCEDURE oder ENTER-PROCEDURE.

Banddateien:
Der Kennwortschutz wird im HDR3-Kennsatz vermerkt

Verschlüsselte Dateien:
Alle EXPASS-Angaben werden wie *UNCHANGED behandelt.

Voreinstellung – nur bei STATE=*NEW: EXPASS=*NONE

= *NONE
Es wird kein Ausführungskennwort vereinbart oder ein bestehendes gelöscht.

= *UNCHANGED
Bei gleichzeitiger Angabe von STATE=*UPDATE bleibt der Wert für EXPASS unverändert, bei gleichzeitiger Angabe von STATE=*NEW wird der Wert EXPASS=*NONE eingetragen.
Falls PROTECT=*BY_DEF_PROT_OR_STD oder STATE=*NEW ohne Angabe zu PROTECT angegeben ist, unterdrückt der Wert *UNCHANGED die Übernahme des entsprechenden von der Funktion „Default-Protection“ gelieferten Wertes.

Bei STATE=*UPDATE gilt: Falls nicht PROTECT=*BY_DEF_PROT_OR_STD angegeben ist, ist die Angabe *UNCHANGED gleichbedeutend mit fehlender Angabe.

Bei STATE=*NEW gilt: Die Angabe *UNCHANGED ist gleichbedeutend mit der Angabe *NONE (unabhängig von der Angabe bei PROTECT).

= <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
definiert ein für den Aufruf des Programms/der Prozedur erforderliches Kennwort.

Die Angabe EXPASS=X'00000000' wird wie *NONE behandelt.

Vor Ausführungszugriff auf die Datei muss „kennwort“ mit dem ADD-PASSWORD-Kommando in die auftragsbezogene Kennworttabelle eingetragen werden (siehe Kommando ADD-PASSWORD im Handbuch „Kommandos“ [3]).
Wird Kennwortvergabe protokolliert, werden die Kennwörter nicht im Klartext ausgegeben.

= (<reg: A(char:4)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 4 Byte, in dem das Ausführungskennwort abgelegt ist.

= <var: char:4>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 4 Byte, in dem das Ausführungskennwort abgelegt ist.

FILE
Pfadname des Katalogeintrags der Datei, Dateigeneration oder Dateigenerationsgruppe, auf den sich die Angaben in den übrigen Operanden des Makroaufrufs beziehen.
- Mit STATE=*NEW (Standard) wird ein Katalogeintrag unter dem angegebenen Namen erstellt.
- Mit STATE=*FOREIGN wird ein auf der mit den Operanden VOLUME und DEVICE angegebenen Privatplatte existierender Katalogeintrag einer Dateigenerationsgruppe importiert.
- Mit STATE=*UPDATE kann ein unter dem angegebenen Pfadnamen existierender Katalogeintrag modifiziert werden; bzw. bei Angabe von Wildcards können alle damit selektierten Katalogeinträge modifiziert werden.

= <c-string 1..80: filename 1..54 with-wild(80)>
Der Pfadname besteht aus [:catid1:][$userid1.]<dateiname1>.

catid$_1$
Kennung des Katalogs, in dem der Katalogeintrag der Datei liegt oder angelegt werden soll. Bei STATE=*UPDATE dürfen Wildcards enthalten sein. Es werden dann jedoch nur Dateien in Katalogen selektiert, die lokal verfügbar sind.
Ist keine Katalogkennung angegeben, wird die voreingestellte Katalogkennung (DEFAULT-PUBSET) des Aufrufers angenommen.

userid$_1$
Benutzerkennung, unter der die Datei liegt oder angelegt werden soll. Nur der Systemverwalter darf bei STATE=*UPDATE auch Wildcards angeben.
- Ist keine Benutzerkennung angegeben, wird die LOGON-Kennung angenommen.
- Der Systemverwalter darf eine fremde Benutzerkennung angeben, wenn für die Datei keine TSOS-Einschränkung vereinbart (siehe Handbuch „SECOS“ [8]) oder nicht STATE=*UPDATE angegeben ist.
- Ein nichtprivilegierter Benutzer darf eine fremde Benutzerkennung angeben, wenn er Mit-Eigentümer der Datei ist.

dateiname$_1$
Name der permanenten oder temporären Datei, der Dateigeneration oder Dateigenerationsgruppe. Bei STATE=*UPDATE darf der Dateiname Wildcards enthalten oder teilqualifiziert sein, d.h. mit einem Punkt enden.

= (<reg: A(char:80)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 80 Byte, in dem der Pfadname abgelegt ist. Belegt der Pfadname nicht die Maximallänge von 80 Byte, muss er mit mindestens einem Blank (X'40') abgeschlossen sein.

= <var: char:80>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 80 Byte, in dem der Pfadname abgelegt ist. Belegt der Pfadname nicht die Maximallänge von 80 Byte, muss er mit mindestens einem Blank (X'40') abgeschlossen sein.

FIRST = zahl
Nur für Dateigenerationsgruppen:
Der Operand darf nur mit STATE=*NEW spezifiziert werden.
Er bestimmt die absolute Generationsnummer der ältesten *katalogisierten* Dateigeneration.
Der Operand ist erforderlich, um den Indexeintrag einer Dateigenerationsgruppe auf privaten Datenträgern (FOREIGN-Dateigenerationsgruppe) zu rekonstruieren. Er bestimmt die Nummer der ältesten zu importierenden Dateigeneration. Er sollte nur zu diesem Zweck benutzt werden.
Auf Band gespeicherte Generationen müssen einzeln mit FILE (Operand STATE= FOREIGN) katalogisiert werden.
Generationen auf Privatplatte können mit IMPORT oder einzeln über FILE (Operand STATE=FOREIGN) importiert werden.

= <integer 1..9999>
Die mit FIRST angegebene Dateigeneration kann nur importiert, nicht jedoch neu katalogisiert werden. Das bedeutet, wenn der erstellte Indexeintrag nicht zur Rekonstruktion einer Dateigenerationsgruppe verwendet werden soll, besteht keine Möglichkeit die hier angegebene Dateigeneration (wie evtl. weitere Dateigenerationen ≤ LAST bzw. ≤ BASE, siehe jeweils dort) auch real anzulegen.

= (<reg: int:2>)
Nur mit MF=M möglich:
Das angegebene Register enthält im unteren Halbwort die Generationsnummer der ältesten katalogisierten Dateigeneration.

= <var: char:2>
Nur mit MF=M möglich:
Symbolische Adresse eines Halbwortes, in dem die Generationsnummer der ältesten katalogisierten Dateigeneration abgelegt ist.

GEN = zahl
Nur für Dateigenerationsgruppen:
legt fest, wie viele Generationen einer Dateigenerationsgruppe höchstens gleichzeitig katalogisiert sein dürfen (vgl. Operand DISP).

Die Angabe GEN kann sich sowohl auf eine neue (STATE=*NEW) als auch auf eine existente Dateigenerationsgruppe (STATE=*UPDATE) beziehen.

Voreinstellung: GEN = 0

= <integer 0..255>
Maximale Anzahl gleichzeitig katalogisierter Dateigenerationen.
Wird GEN=0 mit STATE=*NEW angegeben, dann wird keine Dateigenerationsgruppe, sondern eine „normale“ Datei angelegt; zusammen mit STATE=*UPDATE wird GEN=0 ignoriert.

= (<reg: int:2>)
Nur mit MF=M möglich:
Das angegebene Register enthält im unteren Halbwort die maximale Anzahl der gleichzeitig katalogisierten Dateigenerationen.

= <var: char:2>
Nur mit MF=M möglich:
Symbolische Adresse eines Halbwortes, in dem die maximale Anzahl gleichzeitig katalogisierter Dateigenerationen abgelegt ist.

GROUPAR
Nur für Dateien auf gemeinschaftlichen Platten und auf Net-Storage:
aktiviert die Zugriffskontrolle über BASIC-ACL und legt fest, wie ein Anwender auf die Datei zugreifen darf, der nicht ihr Eigentümer ist, jedoch zur Benutzergruppe des Dateieigentümers gehört, wenn kein GUARDS-Schutz aktiv ist.
Benutzergruppen können nur bei Einsatz des Softwareproduktes SECOS in einem System definiert werden (siehe Handbuch „SECOS“ [8]). In einem System ohne Benutzergruppen und ohne Installation von SECOS gilt der Wert für GROUPAR für alle Anwender außer dem Dateieigentümer (und der Systemverwaltung). Wenn Benutzergruppen im System eingerichtet werden, wird er für die Mitglieder der Benutzergruppe des Dateieigentümers ausgewertet.

Der Operand darf nicht zusammen mit dem Operanden BASACL angegeben werden.

= *NO-ACCESS
Für die Benutzergruppe ist kein Zugriff auf die Datei erlaubt.

= ([READ = *NO / READ = *YES / R = *N / R = *Y]
[,WRITE = *NO / WRITE = *YES / W = *N / W = *Y]
[,EXEC = *NO / EXEC = *YES / X= *N / X = *Y])
Die in der Liste mit *YES bzw. *Y angegebenen Zugriffsarten sind gestattet. Die runden Klammern sind Bestandteil der Zugriffsliste und müssen mit angegeben werden.

Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=NO bzw. R=N	Lesezugriff ist verboten (Voreinstellung).
READ=YES bzw. R=Y	Lesezugriff ist erlaubt. Anders als bei der Zugriffskontrolle über den ACCESS-Operanden ist damit *nicht* automatisch das Recht verbunden, die Datei auszuführen.
WRITE=NO bzw. W=N	Schreibzugriff ist verboten (Voreinstellung).
WRITE=YES bzw. W=Y	Schreibzugriff ist erlaubt. Anders als bei der Zugriffskontrolle über den ACCESS-Operanden ist damit *nicht* automatisch das Recht verbunden, die Datei zu lesen oder auszuführen.
EXEC=NO bzw. X=N	Ausführen der Datei ist verboten (Voreinstellung).
EXEC=YES bzw. X=Y	Ausführen der Datei ist erlaubt (nicht für Dateigenerationsgruppen).

GUARDS
Aktiviert/Deaktiviert die Zugriffskontrolle über GUARDS (Generally Usable Access contRol aDministration System). Mit GUARDS wird die Datei über ein spezielles Zugriffsprofil geschützt. Dieser Zugriffsschutz wird nur wirksam, falls die Funktionseinheit GUARDS des Softwareproduktes SECOS geladen ist (siehe Handbuch „SECOS“ [8]).

Dateischutz über GUARDS wird aktiviert, wenn mindestens eine Zugriffsart (READ/WRITE/EXEC) mit einem Guard-Eintrag im Guard-Katalog verknüpft wird. Auch die Angabe READ/WRITE/EXEC=*NONE ist in diesem Sinne ein Guard-Eintrag und aktiviert den Dateischutz über GUARDS (lesender, schreibender, ausführender Zugriff auf die Datei ist nicht möglich).
Das Zugriffsprofil muss zu diesem Zeitpunkt noch nicht definiert sein; es kann sogar ohne Einsatz von GUARDS zugewiesen werden. In beiden Fällen wird jeder Dateizugriff abgewiesen.

Erst zum Zeitpunkt des Zugriffs auf eine mit GUARDS geschützte Datei wird geprüft, ob der angegebene Guard-Eintrag (Guard-Name) existiert, ob er verwendet werden darf und ob dem Benutzer auf Grund dieses Zugriffsprofils ein Zugriff in der entsprechenden Zugriffsart erlaubt ist.

Hinweise

– Auf eine Datei kann nicht zugegriffen werden, wenn im Dateikatalog Schutz über GUARDS eingetragen ist, für den angegebenen Guard-Namen jedoch noch kein Zugriffsprofil im Guard-Katalog definiert ist.

– Wenn GUARDS-Schutz aktiviert wird, bleibt der vorher festgelegte Zugriffsschutz über BASIC-ACL bzw. USER-ACCESS und ACCESS erhalten.

– Weitere Informationen zum Zugriffsschutz über die Funktionseinheit GUARDS siehe Abschnitt Dateischutz, Handbuch „Einführung in das DVS“ [1].

= *NONE
GUARDS-Schutz wird deaktiviert. Ein bestehender Zugriffsschutz über GUARDS wird zurückgesetzt, die Datei ist nicht mehr über den Schutzmechanismus GUARDS geschützt.

= ([READ=...] [,WRITE=...] [,EXEC=...])
Jede der drei Zugriffsarten (Lesen, Schreiben, Ausführen) kann über einen gesonderten Guard-Eintrag geschützt werden. Wird für eine Datei GUARDS-Schutz aktiviert, werden alle nicht explizit gesetzten Zugriffsarten mit *NONE belegt. Der Zugriff über diese Zugriffsarten ist dann **nicht** möglich.

[READ = *NONE / <c-string: filename 1..18 without cat-gen-vers> / <var: char:18> / (<reg: A(char:18)>)]
Leseschutz über GUARDS aktivieren.

Voreinstellung: READ = *NONE, falls GUARDS-Schutz über eine andere Zugriffsart aktiviert wurde

= *NONE
Der GUARDS-Schutz für lesenden Zugriff wird zurückgesetzt (die Verknüpfung Leseschutz – Zugriffsprofil wird aufgehoben). Die Datei kann nicht gelesen werden. Dateischutz über GUARDS bleibt aktiv.

= <c-string: filename 1..18 without cat-gen-vers>
Name des Zugriffprofils (Guard-Eintrag im Guard-Katalog), mit dem Leseschutz über GUARDS festgelegt wird. Die Datei kann nur gelesen werden, wenn die im Zugriffsprofil festgelegten Bedingungen erfüllt sind.
Der Name kann max. 8 Zeichen lang sein, mit Angabe der Benutzerkennung max. 18 Zeichen lang. Eine Katalogkennung kann nicht angegeben werden.

= (<reg: A(char:18)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 18 Byte, in dem der Name des READ-GUARD abgelegt ist.

= <var: char:18>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 18 Byte, in dem der Name des READ-GUARD abgelegt ist.

[,WRITE = *NONE / <c-string: filename 1..18 without cat-gen-vers> / <var: char:18> / (<reg: A(char:18)>)]
Schreibschutz über GUARDS aktivieren.

Hinweis
Anders als beim Kennwortschutz impliziert das Schreibrecht nicht das Leserecht.

Voreinstellung: READ = *NONE, falls GUARDS-Schutz über eine andere Zugriffsart aktiviert wurde

= *NONE
Der GUARDS-Schutz für schreibenden Zugriff wird zurückgesetzt. (Die Verknüpfung Schreibschutz – Zugriffsprofil wird aufgehoben). Schreibender Dateizugriff ist nicht möglich.
Dateischutz über GUARDS bleibt aktiv.

= <c-string: filename 1..18 without cat-gen-vers>
Name des Zugriffprofils (Guard-Eintrag im Guard-Katalog), mit dem Schreibschutz über GUARDS festgelegt wird.
Schreibender Dateizugriff ist nur möglich, wenn die im Zugriffsprofil festgelegten Bedingungen erfüllt sind.
Der Name kann max. 8 Zeichen lang sein, mit Angabe der Benutzerkennung max. 18 Zeichen lang. Eine Katalogkennung kann nicht angegeben werden.

= (<reg: A(char:18)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 18 Byte, in dem der Name des WRITE-GUARD abgelegt ist.

= <var: char:18>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 18 Byte, in dem der Name des WRITE-GUARD abgelegt ist.

[,EXEC = *NONE / <c-string: filename 1..18 without cat-gen-vers> / <var: char:18> / (<reg: A(char:18)>) / *UNCHANGED])
Ausführungsschutz über GUARDS aktivieren.

Voreinstellung: EXEC = *NONE, falls GUARDS-Schutz über eine andere Zugriffsart aktiviert wurde

= *NONE
Der GUARDS-Schutz für ausführenden Zugriff wird zurückgesetzt (die Verknüpfung Ausführungsschutz – Zugriffsprofil wird aufgehoben). Ausführender Dateizugriff ist nicht möglich. (Der Dateischutz über GUARDS bleibt aktiv).

= <c-string: filename 1..18 without cat-gen-vers>
Name des Zugriffsprofils (Guard-Eintrag im Guard-Katalog), mit dem Ausführungsschutz über GUARDS festgelegt wird.
Ausführender Dateizugriff ist nur möglich, wenn die im Zugriffsprofil festgelegten Bedingungen erfüllt sind.
Der Name kann max. 8 Zeichen lang sein, mit Angabe der Benutzerkennung max. 18 Zeichen lang. Eine Katalogkennung kann nicht angegeben werden.

= (<reg: A(char:18)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 18 Byte, in dem der Name des EXEC-GUARD abgelegt ist.

= <var: char:18>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 18 Byte, in dem der Name des EXEC-GUARD abgelegt ist.

= *UNCHANGED
Nur relevant im Zusammenhang mit der Angabe von PROTECT:
Bei gleichzeitiger Angabe von STATE=*UPDATE bleibt der Wert für GUARDS unverändert, bei gleichzeitiger Angabe von STATE=*NEW wird der Wert GUARDS=*NONE eingetragen.

Der Wert *UNCHANGED unterdrückt insbesondere bei
– PROTECT=*FROM_FILE die Übernahme des entsprechenden Wertes aus der Referenzdatei
– PROTECT=*BY_DEF_PROT_OR_STD und bei STATE=*NEW ohne Angabe zu PROTECT die Übernahme des entsprechenden von der Funktion „Default-Protection“ gelieferten Wertes
– gleichzeitiger Angabe von PROTECT=*STD und STATE=*UPDATE das Zurücksetzen des Wertes im Katalogeintrag auf den Wert GUARDS=*NONE

Bei STATE=*UPDATE gilt: Falls PROTECT nicht angegeben ist, ist die Angabe von *UNCHANGED gleichbedeutend mit fehlender Angabe.

Bei STATE=*NEW gilt: Die Angabe *UNCHANGED ist gleichbedeutend mit der Angabe GUARDS=*NONE (unabhängig von der Angabe bei PROTECT).

IOPERF

Nur für Dateien und Dateigenerationen auf gemeinschaftlichen Datenträgern sowie Dateien auf Net-Storage-Volumes:

Gefordertes Performance-Attribut der Datei für die I/O-Verarbeitung. Es gibt die Performance-Attribute „*VERY-HIGH“, „*STD“ und „*HIGH“. Der höchste zulässige Wert ist benutzerkennungsspezifisch.

Wird eine höhere Performance gefordert, als der Maximalwert es erlaubt, dann wird der Maximalwert in den Katalogeintrag übernommen (kein Returncode!).

Der Operand wird ignoriert (kein Returncode!), wenn:
– er für Dateien spezifiziert wird, die nicht auf einem gemeinschaftlichen Datenträger liegen bzw. angelegt werden sollen.
– die Dateikennung auf dem entsprechenden SM-Pubset eine Default-Storage-Klasse besitzt und physikalische Allokierung verboten ist.

In SM-Pubsets wird durch die Angabe des Operanden eine evtl. im Katalogeintrag der Datei eingetragene Storage-Klasse entfernt. Der Operand darf nicht gleichzeitig mit dem Operanden STOCLAS ≠ *NONE angegeben werden.

Voreinstellung – nur bei STATE=*NEW: IOPERF=*STD

= *STD
Die Datei soll nicht über einen Cache bearbeitet werden.

= *HIGH
Die Datei hat hohe Performance-Priorität und soll, wenn möglich, über einen Cache bearbeitet werden.

= *VERY-HIGH
Die Datei hat sehr hohe Performance-Priorität. Wenn möglich, werden alle Seiten permanent im Globalspeicher (GS) gehalten.

= *USER-MAX
Die Datei wird entsprechend dem höchsten Performance-Attribut verarbeitet, das für die Benutzerkennung zulässig ist.
Durch diese Angabe wird sichergestellt, dass ohne Programmänderung auch bei Erweiterung des Wertebereiches oberhalb von *VERY-HIGH stets der Maximalwert verwendet wird.

IOUSAGE
Nur für Dateien und Dateigenerationen auf gemeinschaftlichen Datenträgern sowie Dateien auf Net-Storage-Volumes:
gibt an, auf welche I/O-Operationen sich das Performance-Attribut (Operand IOPERF) der Datei bezieht.

Der Operand wird ignoriert (kein Returncode!), wenn:
– er für Dateien spezifiziert wird, die nicht auf einem gemeinschaftlichen Datenträger liegen bzw. angelegt werden sollen.
– die Dateikennung auf dem entsprechenden SM-Pubset eine Default-Storage-Klasse besitzt und physikalische Allokierung verboten ist.

In SM-Pubsets wird durch die Angabe des Operanden eine evtl. im Katalogeintrag der Datei eingetragene Storage-Klasse entfernt. Der Operand darf nicht gleichzeitig mit dem Operanden STOCLAS ≠ *NONE angegeben werden.

Voreinstellung – nur bei STATE=*NEW: IOUSAGE=*RDWRT

= *RDWRT
Das Performance-Attribut bezieht sich sowohl auf Lese- wie auf Schreiboperationen.

= *WRITE
Das Performance-Attribut bezieht sich nur auf Schreiboperationen.

= *READ
Das Performance-Attribut bezieht sich nur auf Lese-Operationen.

LARGE
Nur für Dateien/FGG auf Platten:
LARGE bezieht sich wie BACKUP auf die Dateisicherung mit ARCHIVE bzw. HSMS und legt fest, ob bei der automatischen Sicherung die Datei bzw. die Generationen jedes Mal vollständig gesichert werden oder nur die seit der letzten Sicherung veränderten Blöcke.

Voreinstellung – nur bei STATE=*NEW: LARGE=*NO

= *NO
vollständige Sicherung

= *YES
partielle Sicherung: nur die geänderten Blöcke werden gesichert. Diese Angabe ist sinnvoll für große Dateien.

LAST
Nur für Dateigenerationsgruppen:
Der Operand darf nur zusammen mit STATE=*NEW und dem FIRST-Operanden angegeben werden. Er bestimmt die absolute Generationsnummer der jüngsten katalogisierten Dateigeneration.
Der Operand ist erforderlich um den Indexeintrag einer Dateigenerationsgruppe auf privaten Datenträgern zu rekonstruieren und bestimmt die Nummer der jüngsten Dateigeneration, die importiert werden soll.

= <integer 1..9999>
Die mit LAST angegebene Dateigeneration kann nur importiert, nicht jedoch neu katalogisiert werden.
Das bedeutet, wenn der erstellte Indexeintrag nicht zur Rekonstruktion einer Dateigenerationsgruppe verwendet werden soll, besteht keine Möglichkeit die hier angegebene Dateigeneration (wie eventuell weitere Dateigenerationen ≥ FIRST, s. Operand FIRST) auch real anzulegen.

= (<reg: int:2>)
Nur mit MF=M möglich:
Das angegebene Register enthält im unteren Halbwort die Generationsnummer der jüngsten katalogisierten Dateigeneration.

= <var: char:2>
Nur mit MF=M möglich:
Symbolische Adresse eines Halbwortes, in dem die Generationsnummer der jüngsten katalogisierten Dateigeneration abgelegt ist.

LIST
Bestimmt, ob für die verarbeiteten Dateinamen ein Protokoll nach SYSOUT geschrieben werden soll.

= *NO
Es soll nicht protokolliert werden.

= *SYSOUT
Jeder abgearbeitete Dateiname und eventuelle Fehler werden in einer Meldung protokolliert.

= *ERRORS-TO-SYSOUT
Nur Dateinamen, deren Abarbeitung zu Fehlern führte, werden in einer Meldung protokolliert.

MACID
wird nur in Verbindung mit MF=C/D/M ausgewertet und legt jeweils das zweite bis einschließlich dritte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: MACID = DK

= <macid>
ein oder zwei Zeichen lange Zeichenfolge, die jeweils das zweite bis dritte Zeichen der generierten Feldnamen und Equates festlegt.

MANCLAS
Nur für permanente Dateien und Dateigenerationsgruppen auf öffentlichen Datenträgern in SM-Pubsets sowie für Dateien auf Net-Storage eines SM-Pubsets, wenn das Softwareprodukt HSMS geladen ist:
Gibt an, ob die Dateisicherung und -verdrängung über eine Management-Klasse gesteuert werden soll (Näheres siehe Handbuch „HSMS“ [10]).

= *NONE
Es wird keine Management-Klasse zugewiesen. Für die Dateisicherung und -verdrängung sind nur die Angaben der entsprechenden Operanden relevant.

= <c-string: structured-name 1..8>
Die Dateisicherung und -verdrängung wird über die angegebene Management-Klasse gesteuert.
Die Management-Klasse muss existieren und der Anwender muss das Recht haben, sie zu benutzen.
Für Dateien in SF-Pubsets oder auf privaten Datenträgern wird die Angabe ignoriert, für temporäre Dateien abgewiesen.

= (<reg: A(char:8)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 8 Byte, in dem der Name der Management-Klasse abgelegt ist.

= <var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 8 Byte, in dem der Name der Management-Klasse abgelegt ist.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben. In allen Makroaufrufen, die sich durch den MF-Operanden unterscheiden (MF=L/E/D/C/M), muss der Versionsoperand den gleichen Wert haben.

MIGRATE

Nur für Dateien auf gemeinschaftlichen Platten und für Dateien auf Net-Storage relevant:
wird durch das Softwareprodukt HSMS (Hierarchisches Speicher Management System) ausgewertet. Der Anwender kann mit MIGRATE festlegen, ob Dateien, auf die er längere Zeit nicht zugegriffen hat, auf eine Speicherebene mit langsamerem Zugriff verdrängt werden dürfen oder nicht. Die Dateien werden von der Online-Verarbeitungsebene S0 auf die online-verfügbare Hintergrundebene S1 oder die offline-verfügbare Hintergrundebene S2 (z.B. Band) verdrängt (Näheres siehe Handbuch „HSMS“ [10]).

Dateigenerationsgruppen:
der für einen FGG-Index angegebene MIGRATE-Wert ist repräsentativ für die ganze Gruppe.

Voreinstellung:
MIGRATE = *ALLOWED für permanente Dateien
MIGRATE = *INHIBITED für temporäre Dateien

= *ALLOWED
Die Datei darf von S0 auf die Speicherebene S1 oder S2 verdrängt werden.

= *INHIBITED
Die Datei soll nicht verdrängt werden, darf jedoch temporär, z.B. zu Reorganisationszwecken auf eine Hintergrundebene ausgelagert werden.

= *FORBIDDEN
Nur für Benutzer mit dem Recht zur physikalischen Allokierung:
Die Datei darf nicht auf eine Hintergrundebene verdrängt werden.
Diese Angabe wird abgewiesen, wenn die Datei auf eine Hintergrundebene verdrängt ist oder keine Berechtigung zur physikalischen Allokierung besteht.

NEWNAME

*Nur mit STATE=*UPDATE erlaubt:*
Die im Operanden FILE angegebene Datei wird in diesen Namen unbenannt. Dabei ist weder ein Wechsel des Pubsets (Katalog) noch der Benutzerkennung möglich. Eine Datei auf einem Net-Storage-Volume kann nicht in eine temporäre Datei und nicht in eine Dateigeneration umbenannt werden. Siehe auch die Abschnitte „Temporäre Dateien“ auf Seite 132 und „Dateigenerationsgruppen (FGG)“ auf Seite 133.

= <c-string 1..80: filename 1..54 with-wild(80)>
Der Pfadname besteht aus [:$catid_2$:][$userid_2$.]<$dateiname_2$>.

$catid_2$
Kennung des Katalogs, in dem der Katalogeintrag der Datei liegt. Wenn hier eine Angabe erfolgt, dann muss diese identisch mit der bei FILE angegebenen $catid_1$ sein.

userid$_2$
Benutzerkennung, unter der die Datei liegt. Wenn hier eine Angabe erfolgt, dann muss diese identisch mit der bei FILE angegebenen userid$_1$ sein.

dateiname$_2$
Dateiname, in den die im Operanden FILE angegebene Datei dateiname$_1$ umbenannt werden soll. Sind in dateiname$_1$ Wildcards enthalten, kann eine geeignete Konstruktionsangabe angegeben werden. Endet dateiname$_1$ mit einem Punkt, dann darf auch dateiname$_2$ mit einem Punkt enden.
dateiname$_2$ muss angegeben werden, wenn eine Datei/FGG umbenannt werden soll. „dateiname$_1$“ und „dateiname$_2$“ dürfen nicht identisch sein.

Bei Banddateien muss sich dateiname2 von dateiname1 durch die hinzugefügte oder geänderte Versionsbezeichnung unterscheiden.

Bei Einsatz von HSMS können Dateien, die auf die Speicherebene S1 bzw. S2 verdrängt wurden, nicht umbenannt werden.

= (<reg: A(char:80)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 80 Byte, in dem der neue Pfadname abgelegt ist. Belegt der Pfadname nicht die Maximallänge von 80 Byte, muss er mit mindestens einem Blank (X'40') abgeschlossen sein.

= <var: char:80>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 80 Byte, in dem der neue Pfadname abgelegt ist. Belegt der Pfadname nicht die Maximallänge von 80 Byte, muss er mit mindestens einem Blank (X'40') abgeschlossen sein.

OPNBACK
ist speziell für Datenbankdateien (UDS-Dateien) bestimmt. Der Operand ermöglicht es dem Anwender, die Datei auch im geöffneten Zustand durch ARCHIVE sichern zu lassen (siehe Handbuch „ARCHIVE“ [9]). Dabei kann es zu Inkonsistenzen in der Datei kommen; es liegt in der Verantwortung des Anwenders, dies zu vermeiden.

Voreinstellung – nur bei STATE=*NEW: OPNBACK=*NO

= *NO
Nur die geschlossene Datei wird gesichert.

= *YES
Die Datei wird auch gesichert, wenn sie geöffnet ist.

OTHERAR
Nur für Dateien auf gemeinschaftlichen Platten und für Dateien auf Net-Storage:
aktiviert die Zugriffskontrolle über BASIC-ACL und legt fest, wie ein Anwender auf die Datei zugreifen darf, der weder ihr Eigentümer ist noch zur Benutzergruppe des Dateieigentümers gehört, wenn kein GUARDS-Schutz aktiv ist. Benutzergruppen können nur bei Einsatz des Softwareproduktes SECOS in einem System definiert werden (siehe Handbuch „SECOS“ [8]). In einem System ohne Benutzergruppen und ohne Installation von SECOS gilt der Wert für OTHERAR für alle Kennungen außer der des Dateieigentümers. Der Operand darf nicht zusammen mit dem Operanden BASACL angegeben werden.

= *NO-ACCESS
Für die Benutzergruppe ist kein Zugriff auf die Datei erlaubt.

= ([READ = *NO / READ = *YES / R = *N / R = *Y]
[,WRITE = *NO / WRITE = *YES / W = *N / W = *Y]
[,EXEC = *NO / EXEC = *YES / X= *N / X = *Y])
Die in der Liste mit *YES bzw. *Y angegebenen Zugriffsarten sind gestattet. Die runden Klammern sind Bestandteil der Zugriffsliste und müssen mit angegeben werden.

Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=NO bzw. R=N	Lesezugriff ist verboten (Voreinstellung).
READ=YES bzw. R=Y	Lesezugriff ist erlaubt. Anders als bei der Zugriffskontrolle über den ACCESS-Operanden ist damit *nicht* automatisch das Recht verbunden, die Datei auszuführen.
WRITE=NO bzw. W=N	Schreibzugriff ist verboten (Voreinstellung).
WRITE=YES bzw. W=Y	Schreibzugriff ist erlaubt. Anders als bei der Zugriffskontrolle über den ACCESS-Operanden ist damit *nicht* automatisch das Recht verbunden, die Datei zu lesen oder auszuführen.
EXEC=NO bzw. X=N	Ausführen der Datei ist verboten (Voreinstellung).
EXEC=YES bzw. X=Y	Ausführen der Datei ist erlaubt (nicht für Dateigenerationsgruppen).

OWNERAR
Nur für Dateien auf gemeinschaftlichen Platten und für Dateien auf Net-Storage:
aktiviert die Zugriffskontrolle über BASIC-ACL und legt fest, wie der Eigentümer (und die Systemverwaltung) auf die Datei zugreifen darf, wenn kein GUARDS-Schutz aktiv ist.

Der Operand darf nicht zusammen mit dem Operanden BASACL angegeben werden.

= *NO-ACCESS
Für die Benutzergruppe ist kein Zugriff auf die Datei erlaubt.

= ([READ = *NO / READ = *YES / R = *N / R = *Y]
[,WRITE = *NO / WRITE = *YES / W = *N / W = *Y]
[,EXEC = *NO / EXEC = *YES / X= *N / X = *Y])
Die in der Liste mit *YES bzw. *Y angegebenen Zugriffsarten sind gestattet. Die runden Klammern sind Bestandteil der Zugriffsliste und müssen mit angegeben werden.

Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=NO bzw. R=N	Lesezugriff ist verboten (Voreinstellung).
READ=YES bzw. R=Y	Lesezugriff ist erlaubt. Anders als bei der Zugriffskontrolle über den ACCESS-Operanden ist damit *nicht* automatisch das Recht verbunden, die Datei auszuführen.
WRITE=NO bzw. W=N	Schreibzugriff ist verboten (Voreinstellung).
WRITE=YES bzw. W=Y	Schreibzugriff ist erlaubt. Anders als bei der Zugriffskontrolle über den ACCESS-Operanden ist damit *nicht* automatisch das Recht verbunden, die Datei zu lesen oder auszuführen.
EXEC=NO bzw. X=N	Ausführen der Datei ist verboten (Voreinstellung).
EXEC=YES bzw. X=Y	Ausführen der Datei ist erlaubt (nicht für Dateigenerationsgruppen).

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

= <name 1..8>
symbolische Adresse (der Name) der Operandenliste.

PREFIX
wird nur in Verbindung mit MF=C/D/M ausgewertet und legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

= I
ist das voreingestellte Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen.

= pre
„pre“ ist ein ein Zeichen langes Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen sollen.

= *
Es wird kein Präfix generiert.

PROTECT
Gibt an, woher die Schutzattribute übernommen werden sollen, die nicht mit den jeweiligen Operanden explizit angegeben werden.
Folgende Schutzattribute (Operanden) können mit PROTECT vergeben werden (abhängig vom Operandenwert):

Zugriffsschutz	**Schutzattribut**	**CATAL-Operand**
Standard-Zugriffskontrolle (Zugriffsart)	ACCESS	ACCESS
Standard-Zugriffskontrolle (Zugriff fremder Benutzer)	USER-ACCESS	SHARE
Einfache Zugriffskontrolliste	BASIC-ACL	BASACL, OWNERAR, GROUPAR, OTHERAR
Zugriffskontrolle über Guards	GUARDS	GUARDS
Kennwörter	READ-PASSWORD, WRITE-PASSWORD, EXEC-PASSWORD	RDPASS, WRPASS, EXPASS
Binär Löschen	DESTROY-BY-DELETE	DESTROY
Freigabesperre für Speicherplatz	SPACE-RELEASE-LOCK	RELSPAC
Freigabedatum zum Löschen	FREE-FOR-DELETION	DELDATE
Schutzfrist	EXPIRATION-DATE	EXDATE oder RETPD

Die Werte für diese Schutzattribute können je nach Wert des Operanden STATE (NEW oder UPDATE) verschiedene Vorbelegungen haben (siehe Tabellen).

Schutzattribute beim Neukatalogisieren von Dateien

PROTECTION-ATTR= **Schutzattribut**	***FROM_FILE**	***STD** (System-Standardwerte)	***BY_DEF_PROT_OR_STD** **Def-Prot. nicht aktiv**	***BY_DEF_PROT_OR_STD** **Default-Protection aktiv**
ACCESS	von der Referendatei übernommener Wert	WRITE		von der Default-Protection gelieferter Wert
USER-ACCESS		OWNER-ONLY		
BASIC-ACL		NONE		
DESTROY-BY-DELETE		NO		
GUARDS		NONE		
SPACE-RELEASE-LOCK		NO		
READ-PASSWORD	NONE			
WRITE-PASSWORD				
EXEC-PASSWORD				
FREE-FOR-DELETION				
AUDIT				

Eine Schutzfrist (EXPIRATION-DATE) kann beim Ersteintrag nicht vergeben werden. Sie ist bei Dateien implizit mit *NONE vorbelegt, bei Dateigenerationsgruppen mit *TODAY.

Schutzattribute beim Ändern von Dateimerkmalen

<table>
<tr><th>PROTECTION-ATTR=

Schutzattribut</th><th>*UNCH</th><th>*FROM_FILE</th><th>*STD
(System-Standardwerte)</th><th>*BY_DEF_PROT_OR_STD
Def-Prot. nicht aktiv</th><th>*BY_DEF_PROT_OR_STD
Default-Protection aktiv</th></tr>
<tr><td>ACCESS</td><td rowspan="11"></td><td rowspan="7">von der Referenzdatei übernommener Wert</td><td>WRITE</td><td></td><td rowspan="11">von der Default-Protection gelieferter Wert</td></tr>
<tr><td>USER-ACCESS</td><td>OWNER-ONLY</td><td></td></tr>
<tr><td>BASIC-ACL</td><td>NONE</td><td></td></tr>
<tr><td>DESTROY-BY-DELETE</td><td>NO</td><td></td></tr>
<tr><td>GUARDS</td><td>NONE</td><td></td></tr>
<tr><td>SPACE-RELEASE-LOCK</td><td>NO</td><td></td></tr>
<tr><td>EXPIRATION-DATE *)</td><td>TODAY</td><td></td></tr>
<tr><td>READ-PASSWORD</td><td colspan="2" rowspan="5">UNCHANGED</td><td rowspan="5">NONE</td></tr>
<tr><td>WRITE-PASSWORD</td></tr>
<tr><td>EXEC-PASSWORD</td></tr>
<tr><td>FREE-FOR-DELETION</td></tr>
<tr><td>AUDIT</td><td colspan="4"></td></tr>
</table>

*) Die Schutzfrist wird nur dann eingetragen, wenn es sich um eine permanente Datei mit Erstellungsdatum oder um eine Dateigenerationsgruppe handelt. Hat die Referenzdatei keine Schutzfrist, wird *TODAY eingetragen.

= *STD
Es werden diese System-Standardwerte gesetzt:

ACCESS	= *WRITE
BASIC-ACL	= *NONE
USER-ACCESS	= *OWNER-ONLY (auch bei Banddateien)
DESTROY	= *NO
SPACE-RELEASE-LOCK	= *NO
GUARDS	= *NONE
EXPIRATION-DATE	= *TODAY (nur bei permanenten Dateien mit Erstellungsdatum und Dateigenerationsgruppen)

Für einzelne Dateigenerationen wird PROTECT=*STD abgewiesen.

= *BY_DEF_PROT_OR_STD
Die Schutzattribute werden abhängig vom Einsatz der Funktion „Default-Protection“ vergeben.
Ist Default-Protection aktiv, liefert sie Werte für alle Schutzattribute, die oben aufgeführt sind, sofern diese nicht explizit angegeben wurden.
Ist Default-Protection nicht aktiv, werden die Schutzattribute wie bei PROTECT=*STD eingetragen. Zusätzlich werden noch folgende System-Standardwerte übernommen:

FREE-FOR-DELETION	= *NONE
READ-PASSWORD	= *NONE
WRITE-PASSWORD	= *NONE
EXEC-PASSWORD	= *NONE

Bei STATE=*NEW ist PROTECT=*BY_DEF_PROT_OR_STD gleichbedeutend mit fehlender Angabe.

= (*FROM_FILE,<c-string: filename 1..54>)
Alle bei *STD genannten Schutzattribute, die der Aufrufer nicht explizit angibt, werden von der Referenzdatei übernommen. Die von der Referenzdatei übernommenen Schutzattribute werden so behandelt, als seien sie explizit angegeben worden.

Ausnahmen:
– Für eine Dateigenerationsgruppe werden Ausführungsrechte nicht abgewiesen, sondern ignoriert.
– Für temporäre Dateien wird EXDATE nicht abgewiesen, sondern ignoriert.

Falls die Referenzdatei keine Schutzfrist hat, wird für Dateigenerationsgruppen und für permanente Dateien mit Erstellungsdatum EXDATE=*TODAY übernommen.

Kennwörter und Freigabedatum der Referenzdatei werden nicht übernommen.
Bei CATAL STATE=*NEW werden sie mit dem Systemstandardwert *NONE belegt (siehe auch Tabelle „Schutzattribute beim Neukatalogisieren von Dateien“ auf Seite 175).
Bei STATE=*UPDATE mit *UNCHANGED (siehe Tabelle „Schutzattribute beim Ändern von Dateimerkmalen“ auf Seite 176).

Die Referenzdatei muss im gleichen Pubset liegen, wie die mit FILE spezifizierte Datei. Ist keine Katalogkennung angegeben, wird der Default-Katalog der Benutzerkennung angenommen. Deshalb muss die Katalogkennung immer angegeben werden, wenn FILE sich nicht auf den Default-Katalog bezieht.

= (*FROM_FILE,(<reg: A(char:54)>))
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 54 Byte, in dem der Pfadname abgelegt ist. Belegt der Pfadname nicht die Maximallänge von 54 Byte, muss er mit mindestens einem Blank (X'40') abgeschlossen sein.

= (*FROM_FILE,<var: char:54>)
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 54 Byte, in dem der Pfadname abgelegt ist. Belegt der Pfadname nicht die Maximallänge von 80 Byte, muss er mit mindestens einem Blank (X'40') abgeschlossen sein.

RDPASS
Der Benutzer kann mit RDPASS ein Lesekennwort festlegen, ändern oder löschen.

Bei temporären Dateien: kein Kennwortschutz möglich.
Bei Banddateien: das Kennwort wird im HDR3-Kennsatz vermerkt
Bei verschlüsselten Dateien: Alle RDPASS-Angaben werden wie *UNCHANGED behandelt.

Voreinstellung – nur bei STATE=*NEW: RDPASS=*NONE

= *NONE
Es wird kein Lesekennwort vergeben oder ein bestehendes gelöscht.

= *UNCHANGED
Bei gleichzeitiger Angabe von STATE=*UPDATE bleibt der Wert für RDPASS unverändert, bei gleichzeitiger Angabe von STATE=*NEW wird der Wert RDPASS=*NONE eingetragen.
Falls PROTECT=*BY_DEF_PROT_OR_STD oder STATE=*NEW ohne Angabe zu PROTECT angegeben ist, unterdrückt der Wert *UNCHANGED die Übernahme des entsprechenden von der Funktion „Default-Protection“ gelieferten Wertes.

Bei STATE=*UPDATE gilt: Falls nicht PROTECT=*BY_DEF_PROT_OR_STD angegeben ist, ist die Angabe *UNCHANGED gleichbedeutend mit fehlender Angabe.

Bei STATE=*NEW gilt: Die Angabe *UNCHANGED ist gleichbedeutend mit der Angabe *NONE (unabhängig von der Angabe bei PROTECT).

= <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
definiert ein für Lesezugriff erforderliches Kennwort.

Ist ein Programm mit Lesekennwort geschützt, wirkt sich dies auch auf die im Hauptspeicher befindliche Phase aus. Das Kommando LOAD-PROGRAM wird abgelehnt, die IDA-Kommandos DISPLAY und AT werden ebenfalls abgewiesen. Ist ein Quellprogramm mit RDPASS geschützt, kann es nicht übersetzt werden.

= (<reg: A(char:4)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 4 Byte, in dem das Lesekennwort abgelegt ist.

= <var: char:4>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 4 Byte, in dem das Lesekennwort abgelegt ist.

RELSPAC
gibt an, ob die Freigabe von Speicherplatz mit dem Kommando MODIFY-FILE-ATTRIBUTES bzw. dem Makroaufruf FILE erlaubt werden soll.

Voreinstellung – nur bei STATE=*NEW: RELSPAC=*ALLOWED

= *ALLOWED
Der Speicherplatz darf freigegeben werden.

= *IGNORED
Die Freigabe von Speicherplatz wird ignoriert.

= *UNCHANGED
Nur relevant im Zusammenhang mit der Angabe von PROTECT:
Bei gleichzeitiger Angabe von STATE=*UPDATE bleibt der Wert für RELSPAC unverändert, bei gleichzeitiger Angabe von STATE=*NEW wird der Wert RELSPAC= *ALLOWED eingetragen.

Der Wert *UNCHANGED unterdrückt insbesondere bei
– PROTECT=*FROM_FILE die Übernahme des entsprechenden Wertes aus der Referenzdatei
– PROTECT=*BY_DEF_PROT_OR_STD und bei STATE=*NEW ohne Angabe zu PROTECT die Übernahme des entsprechenden von der Funktion „Default-Protection” gelieferten Wertes
– gleichzeitiger Angabe von PROTECT=*STD und STATE=*UPDATE das Zurücksetzen des Wertes im Katalogeintrag auf den Wert RELSPAC=*ALLOWED

Bei STATE=*UPDATE gilt: Falls PROTECT nicht angegeben ist, ist die Angabe von *UNCHANGED gleichbedeutend mit fehlender Angabe.

Bei STATE=*NEW gilt: Die Angabe *UNCHANGED ist gleichbedeutend mit der Angabe RELSPAC=*ALLOWED (unabhängig von der Angabe bei PROTECT).

RETPD
Mit RETPD wird eine Schutzfrist (EXPIRATION-DATE) festgelegt, während derer die Datei nicht geändert oder gelöscht werden kann; d.h. sie kann nur gelesen werden („read only“).

Voreinstellung – nur bei STATE=*NEW: RETPD = 0, d.h. die Datei/Generation kann jederzeit geändert/gelöscht werden

Eine Schutzfrist kann nur für existente Dateien vereinbart werden, d.h. die Katalogfelder CRE-DATE und FILE-STRUC müssen einen Wert ≠ NONE anzeigen. Das heißt auch, die CATAL-Operanden RETPD und STATE=*NEW bzw. STATE=*FOREIGN sind nicht kombinierbar (RETPD wird ignoriert).

Die Berechnung des EXPIRATION-DATE aus der Anzahl der angegebenen Tage erfolgt immer auf lokaler Zeitbasis mit dem Tagesdatum und der Uhrzeit 0 Uhr.

Die Schutzfrist kann mit einem erneuten CATAL-Makroaufruf mit RETPD-Angabe aufgehoben oder verändert werden. Nach Ablauf der Schutzfrist ist wieder Schreibzugriff zugelassen.

Eine bestehende Generation einer FGG kann beim Einrichten einer neuen Generation gelöscht werden, auch wenn die Schutzfrist noch nicht abgelaufen ist (siehe Operand DISP).

Eine gleichzeitige Verwendung der Operanden EXDATE und RETPD ist nicht möglich.

Wird der Operand RETPD angegeben, so wird der von der Funktion „Default-Protection“ gelieferte EXDATE-Wert ignoriert.

= <integer 0..32767>
Anzahl der Tage, die die Datei geschützt sein soll.

= (<reg: int:2>)
Nur mit MF=M möglich:
Das angegebene Register enthält im unteren Halbwort die Anzahl der Tage, die die Datei geschützt sein soll.

= <var: char:2>
Nur mit MF=M möglich:
Symbolische Adresse eines Halbwortes, in dem die Anzahl der Tage abgelegt ist, die die Datei geschützt sein soll.

S0MIGR
*Nur relevant bei STATE=*UPDATE für Dateien in SM-Pubsets, die bereits Speicherplatz belegen:*
Legt fest, ob die Datei innerhalb des SM-Pubsets (Speicherhierarchie-Ebene S0) auf ein anderes Volume-Set umallokiert werden darf.

In SM-Pubsets wird durch die Angabe des Operanden eine evtl. im Katalogeintrag der Datei eingetragene Storage-Klasse entfernt. Der Operand darf nicht gleichzeitig mit dem Operanden STOCLAS ≠ *NONE angegeben werden.

= *ALLOWED
Die Datei darf innerhalb des SM-Pubsets umallokiert werden.

Der Wert wird ignoriert für Dateien und Generationen auf Volume-Sets mit permanenter Datenhaltung, wenn die Dateikennung auf dem entsprechenden SM-Pubset eine Default-Storage-Klasse besitzt und physikalische Allokierung verboten ist.

= *FORBIDDEN
Nur für Benutzer mit dem Recht zur physikalischen Allokierung:
Ein automatisches Umallokieren ist nicht erlaubt; die Datei soll auf dem Volume-Set bleiben, auf dem sie derzeit allokiert ist.
Diese Angabe wird in folgenden Fällen abgewiesen:
– es liegt keine Berechtigung zur physikalischen Allokierung vor
– die Datei liegt zwar in einem SM-Pubset, belegt aber noch keinen Speicherplatz
– die Datei ist zwar in einem SM-Pubset katalogisiert, liegt aber auf einem Net-Storage-Volume

SHARE
legt fest, ob Datei/Dateigenerationen unter anderen Benutzerkennungen als der Eigentümerkennung bearbeitet werden dürfen, wenn kein BASIC-ACL- oder GUARDS-Schutz aktiv ist. Die Art des zulässigen Zugriffs wird durch die übrigen Dateischutzmerkmale festgelegt (siehe die Operanden ACCESS, WRPASS usw.).

Voreinstellung – nur bei STATE=*NEW: – SHARE=*NO für Plattendateien,
– SHARE=*YES für Banddateien

Banddateien:
Das DVS übernimmt beim ersten Eröffnen das SHARE-Kennzeichen in den HDR1-Kennsatz („Zugriffsvermerk“).

= *NO
Die Datei ist nicht mehrbenutzbar.
Bei Banddateien: HDR1-Kennsatz: Zugriffsvermerk = 1

= *YES
Dateizugriff wird für beliebige Benutzerkennungen zugelassen, d.h. Datei/Generationen sind mehrbenutzbar.
Bei temporäre Dateien: SHARE=*YES ist nicht zulässig
Bei Banddateien: HDR1-Kennsatz: Zugriffsvermerk = X'40'

= *SPECIAL
Für die Kennung mit dem Privileg `HW-MAINTENANCE` wird Dateizugriff zugelassen. Implizit gilt SHARE=*YES!
Der Wert darf für Dateigenerationsgruppen nicht angegeben werden.

= *UNCHANGED

Nur relevant im Zusammenhang mit der Angabe von PROTECT:
Bei gleichzeitiger Angabe von STATE=*UPDATE bleibt der Wert für SHARE unverändert, bei gleichzeitiger Angabe von STATE=*NEW wird der Wert SHARE=*NO eingetragen.

Der Wert *UNCHANGED unterdrückt insbesondere bei
- PROTECT=*FROM_FILE
  die Übernahme des entsprechenden Wertes aus der Referenzdatei
- PROTECT=*BY_DEF_PROT_OR_STD sowie bei STATE=*NEW ohne Angabe zu PROTECT die Übernahme des entsprechenden von der Funktion „Default-Protection“ gelieferten Wertes
- gleichzeitiger Angabe von PROTECT=*STD und STATE=*UPDATE
  das Zurücksetzen des Wertes im Katalogeintrag auf den Wert SHARE=*NO

Bei STATE=*UPDATE gilt: Falls PROTECT nicht angegeben ist, ist die Angabe von *UNCHANGED gleichbedeutend mit fehlender Angabe.

Bei STATE=*NEW gilt: Die Angabe *UNCHANGED ist gleichbedeutend mit der Angabe SHARE=*NO (unabhängig von der Angabe bei PROTECT).

STATE
gibt an, ob ein neuer Katalogeintrag erstellt, ein bestehender Katalogeintrag verändert oder ein Katalogeintrag importiert werden soll.

= *NEW
Es wird ein Katalogeintrag erstellt.

= *UPDATE
Ein bestehender Katalogeintrag (FILE=...) wird geändert. STATE=*UPDATE muss bei jedem Zugriff auf einen bestehenden Katalogeintrag angegeben werden. Es werden die Merkmale geändert, deren zugehörige Operanden im aktuellen CATAL-Makroaufruf angegeben werden. Besteht Kennwortschutz, muss das für Schreibzugriff notwendige Kennwort in der Kennworttabelle des Auftrags enthalten sein.

= *FOREIGN

Nur für exportierte FGG auf Privatplatten:
Ein nur im F1-Etikett einer Privatplatte stehender Gruppeneintrag einer auf Privatplatten abgelegten Dateigenerationsgruppe soll importiert werden. Es müssen zusätzlich die Operanden VOLUME und DEVICE angegeben werden; weitere Operanden werden ignoriert oder abgewiesen. Die zu übernehmenden Generationen müssen anschließend einzeln mit dem Makro FILE (Operanden STATE=*FOREIGN, DEVICE und VOLUME) oder zu mehreren mit IMPORT importiert werden.

STOCLAS

Der Operand ist nur relevant für Dateien, Dateigenerationsgruppen und Dateigenerationen auf öffentlichen Datenträgern in SM-Pubsets und Dateien auf einem Net-Storage-Volume, die in einem SM-Pubset katalogisiert sind.

Bei STATE=*UPDATE für Dateien und Dateigenerationen, denen bereits Speicherplatz zugewiesen wurde, gibt er an, ob die Wahl des Ablageortes der Daten (Volume-Set) innerhalb des SM-Pubsets über eine Storage-Klasse gesteuert werden soll.
Die Vergabe einer Storage-Klasse, d.h. die Angabe des Operanden STOCLAS ≠ *NONE, darf nicht gleichzeitig mit der Vergabe der darin enthaltenen Einzelattribute (Operanden AVAIL, DISKWR, IOPERF, IOUSAGE) oder zusammen mit dem Operanden S0MIGR erfolgen.
Der Auftrag wird ebenfalls abgewiesen, wenn die Datei noch keinen Speicherplatz belegt oder wenn die Storage-Klasse das Attribut AVAILABILITY=*HIGH enthält und die Datei derzeit auf eine Hintergrundebene ausgelagert ist (S1- oder S2-Migration mit HSMS).

Für Dateigenerationsgruppen legt der Operand die Default-Storage-Klasse fest, die bei der ersten Speicherplatz-Zuweisung für eine Generation verwendet wird, wenn dabei keine explizite Angabe einer Storage-Klasse bzw. eines der Einzelattribute erfolgt.

Bei STATE=*NEW darf die Vergabe einer Storage-Klasse, d.h. die Angabe des Operanden STOCLAS ≠ *NONE, nicht gleichzeitig mit dem Operanden WORKGRP erfolgen.
Bei STATE=*UPDATE kann nur eine Storage-Klasse vergeben werden, deren WORK-FILE-Attribut mit dem WORK-FILE-Attribut der Dateigenerationsgruppe übereinstimmt.
Einer Datei auf einem Net-Storage-Volume kann eine Storage-Klasse zugewiesen werden. Hierbei kann keine Arbeitsdatei und keine Datei mit vorläufigem Dateiformat K entstehen.

Hinweise

Die Angabe eines Wertes ungleich *NONE beim Operanden STOCLAS kann dazu führen, dass die Datei von ihrem bisherigen Volume-Set verlagert (umallokiert) wird auf einen anderen Volume-Set, der besser zur Storage-Klasse passt. Hierbei können folgende Fälle auftreten:

– Wenn die Storage-Klasse AVAILABILITY=*HIGH enthält und der bisherige Volume-Set AVAILABILITY=*STD besitzt, muss die Datei auf einen Volume-Set mit der Eigenschaft AVAILABILITY=*HIGH umallokiert werden. Ist das Umallokieren nicht möglich, wird der CATAL-Aufruf abgewiesen.

– Wenn die Storage-Klasse eine Volume-Set-Liste enthält und die Datei auf keinem Volume-Set der Volume-Set-Liste liegt, wird die Datei nach Möglichkeit auf einen Volume-Set aus der Liste umallokiert. Ist das Umallokieren nicht möglich, wird der CATAL-Aufruf ohne Umallokieren ausgeführt.

Während des Umallokierens ist die Datei gesperrt (geöffnet), d.h. alle Zugriffe auf die Datei bzw. ihren Katalogeintrag werden abgewiesen, also auch nicht in einen Wartezustand versetzt.

= *NONE
Es wird keine Storage-Klasse zugewiesen; für die Wahl des Ablageortes werden die entsprechenden Einzel-Attribute ausgewertet.

Der Wert wird ignoriert für Dateien und Generationen auf Volume-Sets mit permanenter Datenhaltung, wenn die Dateikennung auf dem entsprechenden SM-Pubset eine Default-Storage-Klasse besitzt und physikalische Allokierung verboten ist.

= *UPDATE
Nur relevant für Dateien, denen eine Storage-Klasse zugewiesen ist, deren Attribute geändert wurden:
Die Dateiattribute werden gemäß der zugewiesenen Storage-Klasse geändert. Wenn die Angabe für Dateigenerationsgruppen erfolgt, wird lediglich geprüft, ob das WORK-FILE-Attribut der Storage-Klasse noch mit dem WORK-FILE-Attribut der Dateigenerationsgruppe übereinstimmt. Ist dies nicht der Fall, wird der Auftrag mit Returncode abgewiesen.

= *STD
Für Dateien und Dateigenerationsgruppen wird die Default-Storage-Klasse der Benutzerkennung für das jeweilige Pubset verwendet.
Im Zusammenhang mit Dateigenerationen wird die Default-Storage-Klasse der Dateigenerationsgruppe verwendet, also die Storage-Klasse, die dem Index der Dateigenerationsgruppe zugewiesen wurde.

= <c-string: structured-name 1..8>
Die Wahl des Ablageortes der Datei wird durch die angegebene Storage-Klasse bestimmt.
Die Storage-Klasse muss existieren und der Anwender muss das Recht haben, sie zu benutzen.
War der Datei die angegebene Storage-Klasse bereits zugewiesen, werden die Dateiattribute nicht aktualisiert, d.h. zwischenzeitliche Änderungen der Storage-Klasse werden nicht wirksam; siehe *UPDATE.
Für Dateien in SF-Pubsets oder auf privaten Datenträgern wird die Angabe ignoriert.
Bei STATE=*NEW und für katalogisierte Dateien ohne Speicherplatzzuweisung wird die Angabe abgewiesen.

= (<reg: A(char:8)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 8 Byte, in dem der Name der Storage-Klasse abgelegt ist.

= <var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 8 Byte, in dem der Name der Storage-Klasse abgelegt ist.

TIMBASE
Gibt an, auf welcher Basis die mit den Operanden EXDATE und DELDATE angegebenen absoluten Datumsangaben interpretiert werden sollen.
Der Operand TIMBASE wirkt nicht auf Datumsangaben, die von der Funktion „Default-Protection“ geliefert wurden. Diese beziehen sich stets auf die lokale Zeit.

= *UTC
Absolute Datumsangaben werden auf Basis der Weltzeit UTC (universal time coordinate) interpretiert.

= *LTI
Alle Datumsangaben werden auf Basis der lokalen Zeit LTI (local time) interpretiert.

USRINFO
Trägt eine benutzereigene Metainformation in den Katalogeintrag der Datei ein. Der Eintrag kann maximal 8 Byte beliebigen Inhalts aufnehmen, dessen Bedeutung der Anwender selbst festlegt. Für Dateien auf privaten Datenträgern wird der Operand ignoriert.

= *NONE
Kein Eintrag bzw. der Eintrag wird gelöscht.

= <c-string 1..8>
Die angegebenen Zeichen werden eingetragen.

= (<reg: A(char:8)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 8 Byte, in dem die einzutragende Metainformation abgelegt ist.

= <var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 8 Byte, in dem die einzutragende Metainformation abgelegt ist.

VERSION
Gibt an, welche Version der Parameterliste generiert werden soll. Es sollte immer die neueste Version verwendet werden!
Die Voreinstellung kann nicht explizit angegeben werden!

implizite Voreinstellung: VERSION=0

Es wird das Parameterlistenformat generiert, das vor BS2000 V9.5A unterstützt wurde. Dieses Format unterstützt allerdings auch nur die bis dahin bekannten Parameter.
Die unterstützten Operanden/Operandenwerte können der Tabelle „Versionsunterschiede – VERSION=0/1/2/3“ auf Seite 193 entnommen werden.

= 1
Es wird das Parameterlistenformat generiert, das in BS2000 V9.5 und V10.0 unterstützt wurde.
Dieses Format unterstützt allerdings auch nur die bis dahin bekannten Parameter.
Die unterstützten Operanden/Operandenwerte können der Tabelle „Versionsunterschiede – VERSION=0/1/2/3“ auf Seite 193 entnommen werden.

= 2
Es wird das Parameterlistenformat für die Versionen BS2000/OSD-BC V1.0 und V2.0 generiert.

= 3
Es wird das Parameterlistenformat für Versionen ab BS2000/OSD-BC V3.0 generiert.

Hinweis

Wenn schon bestehende Software neu übersetzt werden soll, die Manipulationen an der generierten Parameterliste vornimmt, muss das alte Format (0) bzw. (1) bzw. (2) angefordert werden. Ansonsten liegt Source-Kompatibilität vor.

VOLUME
Nur für FGG auf Privatplatten:
Gibt die Archivnummer („vsn“) eines privaten Datenträgers (Privatplatte) an.

Die Operanden VOLUME und DEVICE müssen angegeben werden, wenn eine FGG auf Privatplatten neu erstellt oder rekonstruiert wird (STATE=*NEW) oder wenn eine FGG, die bereits auf Privatplatten existiert, importiert werden soll (STATE=*FOREIGN).

Ist das Softwareprodukt MAREN im Einsatz, kann VOLUME auch ohne Angabe von DEVICE angegeben werden.

= <c-string 1..6>
Archivnummer

= (<reg: A(char:6)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 6 Byte, in dem die Archivnummer abgelegt ist.

= <var: char:6>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 6 Byte, in dem die Archivnummer abgelegt ist.

WORKGRP
Nur relevant beim Einrichten einer Dateigenerationsgruppe in SM-Pubsets:
Legt fest, ob die Dateigenerationsgruppe eine permanente oder eine Arbeits-Dateigenerationsgruppe sein soll.
Arbeits-Dateigenerationsgruppen können zu einem von der Systemverwaltung festgelegten Zeitpunkt durch die Systemverwaltung gelöscht werden.

= *YES
Die Dateigenerationsgruppe wird als Arbeits-Dateigenerationsgruppe eingerichtet.

WRPASS
Der Benutzer kann mit WRPASS ein Schreibkennwort definieren, ändern oder löschen.

temporäre Dateien:
kein Kennwortschutz möglich

Banddateien:
der Kennwortschutz wird im HDR3-Kennsatz vermerkt

Voreinstellung – nur bei STATE=*NEW: WRPASS=*NONE

= *NONE
Es wird kein Schreibkennwort vergeben oder ein bestehendes gelöscht.

= *UNCHANGED
Bei gleichzeitiger Angabe von STATE=*UPDATE bleibt der Wert für WRPASS unverändert, bei gleichzeitiger Angabe von STATE=*NEW wird der Wert WRPASS=*NONE eingetragen.

Falls PROTECT=*BY_DEF_PROT_OR_STD oder STATE=*NEW ohne Angabe zu PROTECT angegeben ist, unterdrückt der Wert *UNCHANGED die Übernahme des entsprechenden von der Funktion „Default-Protection“ gelieferten Wertes.

Bei STATE=*UPDATE gilt: Falls nicht PROTECT=*BY_DEF_PROT_OR_STD angegeben ist, ist die Angabe *UNCHANGED gleichbedeutend mit fehlender Angabe.

Bei STATE=*NEW gilt: Die Angabe *UNCHANGED ist gleichbedeutend mit der Angabe *NONE (unabhängig von der Angabe bei PROTECT).

= <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
definiert ein für Schreibzugriff erforderliches Kennwort.

= (<reg: A(char:4)>)
Nur mit MF=M möglich:
Das angegebene Register enthält die Adresse eines Speicherbereiches von 4 Byte, in dem das Schreibkennwort abgelegt ist.

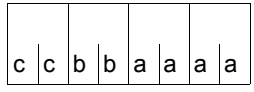
= <var: char:4>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereiches von 4 Byte, in dem das Schreibkennwort abgelegt ist.

Hinweise zur Programmierung

1. Aufruf des CATAL-Makros mit der neuen Operandenliste:
   label CATAL <operanden,...>,VERSION=3
2. Register 1 – Adresse der Operandenliste
3. Der Fehlercode wird nur noch im Standardheader der Parameterliste (Feld IDKRET) und nicht mehr wie in Version 2 im Mehrzweckregister 15 zurückgeliefert.

Returncodes

Standard-header:



Über die Ausführung des Makros CATAL wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Kein Fehler
X'01'	X'00'	X'0000'	Nur mit Kontrolldialogen: Auftrag wurde im Dialog ganz oder teilweise zurückgezogen, d.h. mindestens ein Kontrolldialog wurde mit *NO beantwortet
X'02'	X'00'	X'0000'	Nur im Zusammenhang mit CHECK ≠ *NO: Es ist zwar ein Fehler aufgetreten, aber in einem Fehlerdialog wurde die Fortsetzung der Funktion gefordert
	X'40'	X'0501'	Angeforderter Katalog nicht verfügbar
	X'82'	X'0502'	Angeforderter Katalog im Ruhezustand
	X'40'	X'0503'	Falsche Information im MRSCAT
	X'82'	X'0504'	Fehler im Katalog-Verwaltungs-System
	X'40'	X'0505'	Fehler bei der Rechner-Kommunikation (MRS)
	X'80'	X'0506'	Operation wegen Masterwechsel abgebrochen
	X'40'	X'0510'	Fehler beim Aufruf einer internen Funktion
	X'40'	X'0512'	Angeforderter Katalog unbekannt
	X'40'	X'0513'	Aufruf wurde von System-Exit-Routine abgewiesen
	X'40'	X'051B'	Benutzerkennung im angegebenen Pubset unbekannt
	X'40'	X'051C'	Kein Zugriffsrecht auf angegebenen Pubset
	X'40'	X'051D'	LOGON-Passwort auf angegebenem Pubset anders

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'02'	X'20'	X'0527'	Ein-/Ausgabefehler beim Umallokieren der Daten in einem SM-Pubset
	X'20'	X'0530'	CMS meldet Fehler bei der Speicherplatzanforderung
	X'20'	X'0531'	Unerwarteter Fehler beim Katalogzugriff
	X'82'	X'0532'	Datei gesperrt weil in Gebrauch
	X'82'	X'0534'	Privater Datenträger kann nicht zugewiesen werden
	X'40'	X'0535'	Keine Zugriffsberechtigung auf den Katalogeintrag der Datei (nur im Zusammenhang mit der CCS-Vergabe auf fremder Benutzerkennung)
	X'20'	X'0536'	Fehler im Dateiverwaltungssystem
	X'40'	X'053A'	Fehler beim Ändern des F1-Labels auf privater Platte
	X'20'	X'053B'	Systemfehler beim Dateizugriff
	X'82'	X'053C'	Katalog-Datei des Pubsets ist voll
	X'40'	X'053D'	Katalog oder F1-Etikett-Block ist zerstört
	X'40'	X'053E'	Datei auf privatem Datenträger bereits katalogisiert
	X'82'	X'053F'	Datei ist von einer anderen Task reserviert
	X'40'	X'0540'	Im angegebenen Pubset ist kein Volume-Set verfügbar, das den geforderten Dateiattributen entspricht
	X'82'	X'0541'	Umallokieren der Daten nicht möglich, weil kein geeigneter Volume-Set genügend freien Speicherplatz hat
	X'40'	X'0546'	Katalogeintrag der Datei ist voll
	X'82'	X'054D'	Speicherplatz-Kontingent überschritten
	X'20'	X'054F'	Unerwarteter Fehler beim Zugriff auf JOIN-Datei
	X'40'	X'0555'	STATE=*FOREIGN: Angegebene Datei existiert bereits im Katalog des Benutzers
	X'82'	X'055A'	Geräte zurzeit belegt
	X'40'	X'055C'	Katalogeintrag auf Privatplatte nicht gefunden
	X'40'	X'055D'	Benutzer hat kein Recht zur physikalischen Allokierung
	X'40'	X'055F'	Datenträger konnte nicht belegt werden
	X'01'	X'0576'	Widersprüchliche Operandenkombination oder reservierte Felder des Parameterbereiches verwendet
	X'20'	X'0577'	Interner Fehler beim Zugriff auf die Auftragsumgebung
	X'20'	X'0578'	Interner Fehler bei der Überprüfung der Zugriffsrechte
	X'01'	X'0579'	Ungültiger Operand für temporäre Datei angegeben
	X'40'	X'057A'	Attribut kann für Arbeitsdatei nicht vergeben werden
	X'40'	X'057E'	HSMS nicht verfügbar
	X'40'	X'057F'	Datei ist verdrängt, Umbenennen nicht möglich

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'02'	X'01'	X'0590'	Volume-Angabe ohne Device-Angabe unzulässig
	X'82'	X'0594'	Nicht genug virtueller Speicher verfügbar (auch bei einer Auswahlangabe (Wildcard), wenn zu viele Dateien selektiert wurden)
	X'01'	X'0599'	Operand wird in der RFA-BS-Version nicht unterstützt
	X'40'	X'05A0'	Aktualisieren der Performance-Attribute (DISKWR, IOPERF, IOUSAGE) nicht erlaubt, wenn Daten aus dem Schreibcache noch nicht geschrieben wurden
	X'01'	X'05A8'	Angeforderter Gerätetyp im System nicht gefunden
	X'40'	X'05AD'	Nur beim Umbenennen mit gleichzeitiger S0-Migration: Dateiattribute wurden modifiziert, aber die Datei konnte wegen unerwartetem CMS-Problem nicht umbenannt werden
	X'82'	X'05B0'	Zurzeit kein passendes Gerät verfügbar
	X'40'	X'05B4'	Nur im Zusammenhang mit VOLUME/DEVICE: Eine MOUNT-Meldung für den angeforderten Datenträger wurde vom Operator mit 'NO' beantwortet
	X'40'	X'05B5'	Guard nicht verfügbar
	X'40'	X'05BD'	Unzulässige Kombination von Datei- und Volume-Set-Eigenschaften
	X'20'	X'05C7'	Interner Fehler im DMS
	X'82'	X'05C8'	Maximale Anzahl Dateien für Benutzerkennung erreicht
	X'20'	X'05CA'	Interner Fehler bei Modifikation des CE-Kontingents
	X'01'	X'05CB'	Fehlerhafter/unerlaubter erster Dateiname
	X'40'	X'05CC'	Dateiname bereits katalogisiert
	X'01'	X'05CD'	Fehlerhafter/unerlaubter neuerDateiname
	X'40'	X'05CE'	Erster Dateiname noch nicht katalogisiert
	X'40'	X'05CF'	Datei ist mit Passwort geschützt
	X'82'	X'05D0'	Datei gesperrt weil in Gebrauch
	X'40'	X'05D1'	Fehler bei der Geräteanforderung
	X'40'	X'05D2'	EXPIRATION-DATE wurde für Datei ohne Inhalt angegeben
	X'01'	X'05D3'	GUARD-Name fehlerhaft
	X'40'	X'05D4'	GUARDS-Katalog darf nicht mit GUARD geschützt werden
	X'01'	X'05E8'	Dateiname ungültig für Plattendatei
	X'01'	X'05EE'	Dateiname zu lang
	X'01'	X'05EF'	BASIC-ACL oder GUARD kann nicht vergeben werden
	X'01'	X'05FA'	Zugriff auf REMOTE-IMPORTED-Pubset nicht möglich
	X'40'	X'05FC'	Angegebene Benutzerkennung nicht im Home-Pubset

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'02'	X'40'	X'05FD'	Datei ist durch USER-ACCESS oder EXPIRATION-DATE schreibgeschützt (nur bei CCS-Vergabe auf fremder Benutzerkennung)
	X'40'	X'0606'	Datenträger-Anforderung von MAREN abgewiesen
	X'40'	X'0609'	Aktion für Systemdatei nicht erlaubt
	X'40'	X'060D'	Fehler beim Lesen der Attribute einer Referenzdatei (Op. PROTECT) – Syntaktisch fehlerhafter Dateiname, ggf. auch im Zusammenhang mit einer ACS-Ersetzung – Angegebene Referenzdatei nicht zugreifbar
	X'40'	X'0610'	Mindestens für einen der ausgewählten Dateinamen lieferte die Funktionsausführung einen Returncode
	X'01'	X'0611'	Fehlerhafte Konstruktionsangabe (Oper. NEWNAME mit Wildcards)
	X'40'	X'0613'	Unbekannte Management-Klasse
	X'40'	X'0614'	Keine Zugriffsberechtigung für Management-Klasse
	X'40'	X'0616'	Angegebene Attribute erfordern eine S0-Migration, aber die Datei ist gegen Umallokieren gesperrt
	X'40'	X'0618'	Unbekannte Storage-Klasse
	X'40'	X'0619'	Keine Zugriffsberechtigung für Storage-Klasse
	X'40'	X'0640	Zugriff auf Net-Storage wird vom Subsystem ONETSTOR wegen Kommunikationsproblemen mit dem Net-Client abgewiesen
	X'40'	X'0643'	Net-Client meldet Zugriffsfehler
	X'40'	X'0644'	Net-Client meldet internen Fehler
	X'40'	X'0645'	Datei auf Net-Storage nicht vorhanden
	X'40'	X'0646'	FGG auf Net-Storage-Volume nicht erlaubt
	X'40'	X'0649'	Net-Server meldet POSIX-ACL-FEHLER
	X'40'	X'064A'	Net-Client meldet Zugriff auf Dateien auf dem Net-Storage-Volume verboten
	X'40'	X'064B'	Zugriff auf Node-Files vom Net-Client nicht unterstützt
	X'40'	X'0666'	Datei ist durch ACL oder GUARDS schreibgeschützt (nur bei CCS-Vergabe auf fremder Benutzerkennung)
	X'40'	X'0685'	Datei belegt keinen Speicherplatz und es soll AVAIL=*HIGH, eine Storage-Klasse oder eine S0-Migrationssperre gesetzt werden
	X'20'	X'069D'	Fehlerhaft aufgebauter Katalogeintrag
	X'40'	X'06A6'	AUDIT-Angabe für Benutzerkennung nicht zugelassen
	X'00'	X'06A9'	Einige Generationen der Dateigenerationsgruppe fehlen
	X'40'	X'06B6'	Attribute der Datei passen nicht zur Dateigenerationsgruppe

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'01'	X'06C1'	Mehr als 255 Generationen gefordert oder Konflikt mit BASE-, LAST- oder FIRST-Operanden
	X'01'	X'06C3'	Unzulässiger Name für eine Dateigenerationsgruppe
	X'40'	X'06C4'	Dateigenerationsgruppe noch nicht katalogisiert
	X'01'	X'06C5'	Name für eine Dateigenerationsgruppe zu lang
	X'01'	X'06C6'	Name oder Merkmal einer Banddatei nicht modifizierbar
	X'01'	X'06C7'	Ungültige Generationsnummer angegeben
	X'01'	X'06C8'	Attribut kann nur für die gesamte Dateigenerationsgruppe modifiziert werden
	X'01'	X'06C9'	Generationsspezifische Operanden im falschen Kontext
	X'00'	X'06CA'	Kommando außer fehlerhafter BASE-Angabe ausgeführt
	X'40'	X'06CC'	nur bei Auswahlangabe (Wildcard): Keine Datei entspricht der Auswahlangabe
	X'40'	X'06CD'	Angegebene Dateigenerationsgruppe mit Schreibschutz gegen Erweiterungen gesperrt
	X'01'	X'06CE'	Schutzfrist (RETPD, EXDATE) oder Lösch-Freigabedatum (DELDATE) fehlerhaft angegeben
	X'40'	X'06D5'	Löschen überzähliger Dateigenerationen wird durch Schreibschutz verhindert
	X'01'	X'06DA'	Unzulässige Kombination von privaten und öffentlichen Datenträgern für eine Dateigenerationsgruppe
	X'01'	X'06DB'	Fehlerhafte VOLUME- und DEVICE-Angabe
	X'01'	X'06FA'	Neuer Dateiname nur mit STATE=*UPDATE erlaubt
	X'01'	X'06FB'	Vergabe von Ausführungsrechten für Dateigenerationsgruppe nicht möglich
	X'01'	X'06FD'	Parameterbereich ungültig oder nicht zugreifbar
	X'40'	X'06FF'	BCAM-Verbindung abgebrochen
	X'01'	X'FFFF'	Falsche Funktionsnummer im Parameterbereichs-Header
	X'03'	X'FFFF'	Falsche Versionsnummer im Parameterbereichs-Header

Versionsunterschiede – VERSION=0/1/2/3

MF=	Operand	ohne Vers	Vers=1	Vers=2	Vers=3	Bemerkungen
MF=E		(1)	(1)	(1)	x	
	VERSION		x	x	x	
	PARAM	-	-	-	x	
MF=D/ MF=C		x	x	x	x	
	PREFIX	x	x	x	x	
	MACID	-	-	-	x	
	VERSION	-	x	x	x	
MF=M		-	-	-	x	
	PREFIX	-	-	-	x	
	MACID	-	-	-	x	
	alle Oper. von MF=I/L	-	-	-	x	
MF=I/ MF=L		x	x	x	x	
	$pfadname_1$	(2)	(2)	(2)	-	
	$pfadname_2$	(3)	(3)	(2)	-	
	ACCESS	x	x	x	x	
	ACLPROT	-	(x)	x	x	
	ADMINFO	-	-	-	x	
	AUDIT	x	x	x	x	
	BACKUP	x	x	x	x	
	BASACL	-	(x)	x	x	
	BASE	x	x	x	x	
	CCS	-	-	x	x	
	CHECK	-	-	-	x	
	DELDATE	-	-	-	x	
	DESTROY	x	x	x	x	
	DEVICE	x	x	x	x	(5)
	DISKWR	-	-	x	x	
	DISP	x	x	x	x	
	EXDATE	-	-	-	x	
	EXPASS	x	x	x	x	
	FILE	(2)	(2)	(2)	(4)	

MF=	Operand	ohne Vers	Vers=1	Vers=2	Vers=3	Bemerkungen
MF=I/ MF=L (Forts.)	FIRST	x	x	x	x	
	GEN	x	x	x	x	
	GROUPAR	-	(x)	x	x	
	GUARDS	-	-	x	x	
	IOPERF	-	-	x	x	
	IOUSAGE	-	-	x	x	
	LARGE	x	x	x	x	
	LAST	-	-	x	x	
	LIST	-	-	-	x	
	MANCLAS	-	-	-	x	
	MIGRATE	-	x	x	x	(6)
	NEWNAME	3	2	2	4	
	OPNBACK	-	(x)	x	x	
	OTHERAR	-	(x)	x	x	
	OWNERAR	-	(x)	x	x	
	PROTECT	-	-	-	x	
	RDPASS	x	x	x	x	
	RELSPAC	-	-	x	x	
	RETPD	x	x	x	x	
	SHARE	x	x	x	x	
	STATE	x	x	x	x	
	STOCLAS	-	-	-	x	
	S0MIGR	-	-	-	x	
	TIMBASE	-	-	-	x	
	USRINFO	-	-	-	x	
	VERSION	-	x	x	x	
	VOLUME	x	x	x	x	
	WORKGRP	-	-	-	x	
	WRPASS	x	x	x	x	

Legende

x Operand ist in der Makroversion verfügbar

(x) Operand ist in der Makroversion verfügbar, jedoch nicht ab der ersten Freigabe

\- Operand ist in der Makroversion nicht verfügbar

Vers Version

(1) Format MF=(E,<addr>)

(2) Pfadname mit maximal 54 Zeichen, Format: [:catid:][$userid.]dateiname

(3) Dateiname ohne Catid und Userid (maximal 44 Zeichen)

(4) Auswahl bzw. Konstruktionsangabe (analog 2) mit maximal 80 Zeichen

(5) nur die jeweils unterstützten Gerätetypen

(6) Vers=1: Operandenwert heißt INHIBIT an Stelle von INHIBITED (Vers=2)

In der Tabelle sind unter MF=L die Stellungsoperanden vor den Schlüsselwortoperanden eingeordnet.

CHKFAR – Zugriffsrechte auf Datei überprüfen

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Der Makro CHKFAR überprüft eine im Aufruf angegebene Datei auf ihre Schutzmerkmale und informiert den Aufrufer darüber, welche Zugriffsrechte auf diese Datei er besitzt. Der Anwender kann dabei wählen,

– ob ihm alle Zugriffsmöglichkeiten angezeigt werden, die ihm für diese Datei zur Verfügung stehen (ohne Berücksichtigung eines möglichen Passwortschutzes oder einer Schutzfrist) oder

– ob er Informationen über ein spezielles Zugriffsrecht erhält (unter Einbeziehung eines möglichen Passwortschutzes oder einer Schutzfrist für diese Datei).

Die Informationen werden dem Aufrufer in einem Ausgabebereich der Operandenliste übergeben.

Bei privaten Dateien wertet der Makro CHKFAR nur die Informationen aus dem Benutzerkatalog und nicht aus dem F1-Label aus.

Bei Banddateien wertet der Makro CHKFAR nur die Informationen aus dem Benutzerkatalog und nicht aus dem Header-Satz auf dem Band aus.

Format

Operation	Operanden
CHKFAR	,FILE={'pfadname' | adr1 | (r)} ,ACCESS={*ANY | *READ | *WRITE | *UPDATE | *DELETE | *EXEC | adr2} MF={L | M}
	MF=E,PARAM={adr | (r)}
	MF=D,[,PREFIX=pre]
	MF=C,[,PREFIX=pre][,MACID=macid]

Operandenbeschreibung

FILE
Bezeichnet die Datei, für die der Aufrufer seine Zugriffsrechte erfahren oder überprüfen will.

= pfadname
bezeichnet die Datei, deren Zugriffsrechte überprüft werden sollen, mit:
<c-string 1..54: filename 1..54>

Pfadname bedeutet [:catid:][$userid.]dateiname

catid
Katalogkennung: Falls nicht angegeben, wird die Default-Catid der Benutzerkennung angenommen.

userid
Benutzerkennung: Falls nicht angegeben, wird die Benutzerkennung des SET-LOGON-PARAMETERS- bzw. des LOGON-Kommandos angenommen.

dateiname
vollqualifizierter Dateiname

= adr1
adr1 ist die symbolische Adresse (der Name) eines 54 Byte langen Feldes im Anwenderprogramm, das den Pfadnamen der zu überprüfenden Datei enthält.

= (r)
r ist die Nummer eines Registers, das die Adresse des Feldes adr1 enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

ACCESS
Legt fest, ob sämtliche Zugriffsrechte des Anwenders für die angegebene Datei ausgegeben werden oder ob die Datei auf einzelne Zugriffsrechte des Anwenders hin überprüft wird. Die Informationen werden in einem Ausgabebereich der Operandenliste übergeben.

= *ANY
In einem Ausgabebereich der Operandenliste werden Informationen über sämtliche Zugriffsrechte hinterlegt, die der Aufrufer auf die angegebene Datei hat. Mögliche weitere Schutzmerkmale der Datei wie Passwörter und Schutzfristen werden dabei nicht ausgewertet.

= *READ
Es wird überprüft, ob der Aufrufer die angegebene Datei lesen darf. Dabei wird auch ein möglicher Passwortschutz der Datei berücksichtigt.

= *WRITE
Es wird überprüft, ob der Aufrufer in die angegebene Datei schreiben darf. Dabei wird auch ein möglicher Passwortschutz oder eine Schutzfrist der Datei berücksichtigt.

= *UPDATE
Es wird überprüft, ob der Aufrufer die angegebene Datei lesen und in sie schreiben darf. Dabei wird auch ein möglicher Passwortschutz oder eine Schutzfrist der Datei berücksichtigt.

= *DELETE
Es wird überprüft, ob der Aufrufer die angegebene Datei löschen darf. Dabei wird auch ein möglicher Passwortschutz oder eine Schutzfrist der Datei berücksichtigt.

= *EXEC
Es wird überprüft, ob der Aufrufer die angegebene Datei ausführen darf. Dabei wird auch ein möglicher Passwortschutz der Datei berücksichtigt.

MACID
legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: MACID = RMZ

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

= adr
adr ist die symbolische Adresse (der Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

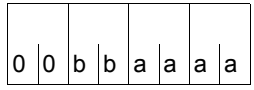
PREFIX
legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: PREFIX = S

= pre
„pre“ ist ein Zeichen langes Präfix, mit dem die generierten Feldnamen und Equates beginnen sollen.

Returncodes

Standard-header:



Über die Ausführung des Makros CHKFAR wird im Standardheader folgender Returncode übergeben (bb = SUBCODE1, aaaa = MAINCODE):

X'bb'	X'aaaa'	Erläuterung
X'00'	X'0000'	Funktion erfolgreich ausgeführt.
X'01'	X'6000'	Die Funktion konnte nicht ausgeführt werden: Ungültiger Wert in der Operandenliste.
X'40'	X'6001'	Die Funktion konnte nicht ausgeführt werden: Die angegebene Datei ist nicht im Katalog verzeichnet.
X'40'	X'6008'	Die Funktion konnte nicht ausgeführt werden: Der angegebene Katalog ist nicht bekannt oder nicht verfügbar.
X'20'	X'6014'	Die Funktion konnte nicht ausgeführt werden: Systemfehler.
X'40'	X'6021'	BCAM-Verbindungsfehler
X'40'	X'6022'	BCAM-Verbindung unterbrochen
X'01'	X'6040'	Die Funktion konnte nicht ausgeführt werden: Die Operandenliste ist nicht in der erforderlichen Länge verfügbar oder zugewiesen.

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Beschreibung der Ausgabefelder

– Bei ACCESS = *ANY werden die folgenden Informationen zurückgegeben:

  Zugriffsrechte: die Zugriffsrechte, die der Aufrufer für die Datei besitzt.
  Passwörter und die Schutzfrist werden nicht berücksichtigt.

  – Ist Zugriffsschutz über GUARDS festgelegt, so werden die Guards für den Zugreifer ausgewertet.

  – Ist eine BASIC-ACL gültig, werden die Werte aus dem für den Aufrufer gültigen Eintrag genommen.

  – Ist eine ACL gültig, wird der Wert „null“ im Feld ACCESS-RIGHTS geliefert.

    Seit SECOS V4.0 wird die Zugriffskontrolle über ACL nicht mehr unterstützt.

  – Ist SHARE/ACCESS gültig, werden die Werte wie folgt gesetzt:

ACCESS	SHARE	Eigent.			Andere		
		R	W	X	R	W	X
WRITE	NO	Y	Y	Y	N	N	N
WRITE	YES	Y	Y	Y	Y	Y	Y
READ	NO	Y	N	Y	N	N	N
READ	YES	Y	N	Y	Y	N	Y

  Legende: R: READ, W: WRITE, X: EXECUTE, Y: YES, N: NO

  Wenn die zu überprüfende Datei existiert, der Aufrufer jedoch keine Zugriffsrechte auf diese Datei besitzt, liefert der Makro CHKFAR den Returncode null und die Zugriffsrechte im Feld ACCESS-RIGHTS.

– Bei ACCESS = *READ/*WRITE/*UPDATE/*DELETE/*EXEC:

  CHECK-RESULT: gibt an, ob der erwünschte Zugriff erlaubt ist oder nicht.

Layout der Operandenliste

(Makroauflösung mit MF=D und Standardwerten für PREFIX und MACID)

```
          CHKFAR MF=D
1         MFCHK MF=D,PREFIX=S,MACID=RMZ,PARAM=,                          C
1               SUPPORT=(C,D,E,L,M),DMACID=RMZ,SVC=8
2 SRMZ    DSECT ,
2               *,##### PREFIX=S, MACID=RMZ #####
1 ***************************************************************
1 *               CHKFAR  - PARAMETERAREA                       *
1 ***************************************************************
1         #INTF REFTYPE=REQUEST,INTNAME=CHKFAR,INTCOMP=001
1 *
1 SRMZPA    DS    OF          BEGIN of PARAMETERAREA          _INOUT
1 *
1         FHDR  MF=(C,SRMZ),EQUATES=NO
2         DS    0A
2 SRMZFHE DS    0XL8          0   GENERAL PARAMETER AREA HEADER
2 *
2 SRMZIFID DS   0A            0   INTERFACE IDENTIFIER
2 SRMZFCTU DS   AL2           0   FUNCTION UNIT NUMBER
2 *                               BIT 15    HEADER FLAG BIT,
2 *                               MUST BE RESET UNTIL FURTHER NOTICE
2 *                               BIT 14-12 UNUSED, MUST BE RESET
2 *                               BIT 11-0  REAL FUNCTION UNIT NUMBER
2 SRMZFCT DS    AL1           2   FUNCTION NUMBER
2 SRMZFCTV DS   AL1           3   FUNCTION INTERFACE VERSION NUMBER
2 *
2 SRMZRET DS    0A            4   GENERAL RETURN CODE
2 SRMZSRET DS   0AL2          4   SUB RETURN CODE
2 SRMZSR2 DS    AL1           4   SUB RETURN CODE 2
2 SRMZSR1 DS    AL1           5   SUB RETURN CODE 1
2 SRMZMRET DS   0AL2          6   MAIN RETURN CODE
2 SRMZMR2 DS    AL1           6   MAIN RETURN CODE 2
2 SRMZMR1 DS    AL1           7   MAIN RETURN CODE 1
2 SRMZFHL EQU   8             8   GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *
1 SRMZACC DS    XL1         ACCESS                                  001
1 SRMZANY EQU   0             = *ANY                                001
1 SRMZREA EQU   1             = *READ                               001
1 SRMZWRI EQU   2             = *WRITE                              001
1 SRMZUPD EQU   3             = *UPDATE                             001
1 SRMZDEL EQU   4             = *DELETE                             001
1 SRMZEXE EQU   5             = *EXEC                               001
1 *
1 SRMZFILE DS   CL54        FILE   = pathname                       001
```

```
1 *
1 SRMZRIF  DS    XL1            RETURN_INFO                          001
1 * Here after: ACCESS_RIGHTS bits returned when ACCESS = *ANY
1 SRMZARR  EQU   X'80'             READ                               001
1 SRMZARW  EQU   X'40'             WRITE                              001
1 SRMZARE  EQU   X'20'             EXEC                               001
1 SRMZARU  EQU   X'1F'             UNUSED                             001
1 * Here after: AUTHORIZATION bin-value returned when ACCESS NE *ANY
1 SRMZALW  EQU   0                 ALLOWED                            001
1 SRMZFBD  EQU   1                 FORBIDDEN                          001
1 *
1 SRMZUNU  DS    XL4            -- MUST BE ZERO --                   001
1 SRMZPA#  EQU   *-SRMZPA       LENGTH OF PARAMETERAREA              001
```

CHNGE – TFT-Eintrag ändern

Makrotyp: S-Typ (E-Form/L-Form) (siehe Seite 870)

Der CHNGE-Makroaufruf ändert den Dateikettungsnamen in einem Eintrag der Task File Table (TFT), d.h.: einer Datei wird ein neuer Dateikettungsname zugewiesen. Alle übrigen Werte in diesem TFT-Eintrag bleiben unverändert.

CHNGE kann nicht auf den TFT-Eintrag einer geöffneten Datei angewendet werden.

Format

Operation	Operanden
CHNGE	[name$_1$],name$_2$[,MF=L]
	MF=(E,{ adr / (r) })

Operandenbeschreibung

Die Formen des MF-Operanden sind detailliert im Anhang, Seite 870 beschrieben.

name$_1$
1-8 Zeichen langer Dateikettungsname, der durch „name$_2$“ ersetzt werden soll.

Voreinstellung: der erste TFT-Eintrag mit dem Dateikettungsnamen C'␣␣␣␣␣␣␣␣' wird bearbeitet (z.B. über LOCK-FILE-LINK-Funktion eingerichtet).

name$_2$
1-8 Zeichen langer Dateikettungsname, der den „name$_1$“ ersetzt.

Hinweis zur Programmierung

Rücksprungkennzeichen werden im Register 15 abgesetzt:

X'00' - die Anforderung wurde ordnungsgemäß abgeschlossen
X'05A6' - Zweiter Operand fehlerhaft
X'05C2' - Dateikettungsname enthält unzulässige binäre Nullen
X'05D5' - Dateikettungsname nicht vorhanden
X'05D6' - Datei mit angegebenem Dateikettungsnamen zurzeit geöffnet
X'05DD' - Zweiter Dateikettungsname bereits vorhanden

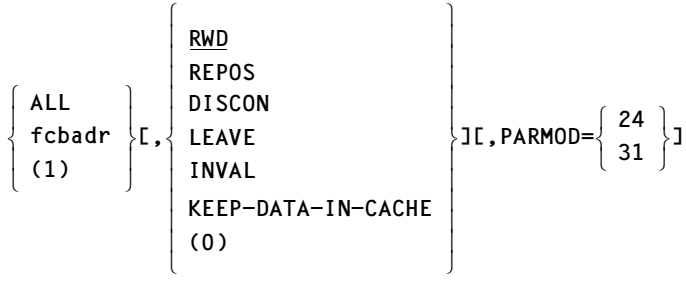
CLOSE – Datei schließen

Der CLOSE-Makroaufruf schließt Dateien, d.h. er trennt sie von dem Benutzerprogramm, in dem sie eröffnet wurden. Alle Ein-/Ausgabepuffer, die das System bei der Dateieröffnung automatisch angelegt hat, werden freigegeben. Der FCB wird mit dem Inhalt versehen, den er vor dem Eröffnen hatte.

Während der CLOSE-Verarbeitung kann das Benutzerprogramm – ähnlich wie bei der OPEN-Verarbeitung – an EXLST-Ausgängen das Band positionieren (CLOSPOS) oder Benutzerkennsätze schreiben (LABEND).

Ein CLOSE-Makroaufruf auf eine nicht geöffnete Datei wird ohne Meldungsausgabe ignoriert.

Format

Operation	Operanden
CLOSE	

Operandenbeschreibung

fcbadr
Adresse des FCB der zu schließenden Datei

(0)
Das Register 0 enthält im rechten Byte den Positionierungsschlüssel bzw. CLOSE-Modus:

X'00'	LEAVE
X'01'	DISCON
X'02'	REPOS
X'03'	RWD
X'05'	INVAL (nur bei PARMOD=31)
X'06'	KEEP-DATA-IN-CACHE

(1)
Register 1 enthält die FCB-Adresse.

ALL
Schließt alle Dateien, die im laufenden Programm geöffnet, aber noch nicht geschlossen wurden. Systemdateien und EAM-Dateien sind vom CLOSE nicht betroffen. Wird eine Datei nicht ordnungsgemäß geschlossen, so erfolgt eine Warnung.

DISCON
Für Banddateien:
das Band wird auf Bandanfang positioniert und entladen bzw. freigegeben. Ein evtl. mit einem FILE-Aufruf reserviertes Gerät bleibt dem Auftrag weiterhin zugeordnet; es wird erst durch einen nachfolgenden RELEASE-Aufruf (REL-Makro) freigegeben.

INVAL
Für Plattendateien:
Die zur Datei im Cache liegenden Seiten sollen nur invalidiert, aber nicht auf die Platte zurückgeschrieben werden, d.h. nach dem CLOSE sind die Daten verloren. INVAL kann nur bei PARMOD=31 angegeben werden.

KEEP-DATA-IN-CACHE
Für Plattendateien:
Die Daten, die in einem Cache zwischengepuffert wurden, werden beim CLOSE nicht auf die Platte gesichert. Ein nachfolgender OPEN auf die gleiche Datei kann dann diese Daten sofort nutzen.

Hinweis

So geschlossene Dateien können mit dem Kommando SHOW-FILE-ATTRIBUTES (Operand CACHE-NOT-SAVED) angezeigt werden. Ein Sicherung der Daten aus dem Cache auf die Platte kann entweder durch einen weiteren OPEN/CLOSE-Zyklus ohne diese Funktion oder implizit während des Cache-Abbaus mit dem Systemverwalterkommando STOP-PUBSET-CACHING oder EXPORT-PUBSET erzwungen werden. Eine dateispezifische Cache-Sicherung für geschlossene Dateien ist nicht möglich.

LEAVE
Für Banddateien:
das Band wird abhängig von der LABEL-Angabe bei FILE oder FCB auf das logische Dateiende positioniert.

Wurde im FILE-Aufruf der Operand BYPASS angegeben, verändert sich die Bandposition nicht, und die CLOSPOS-Routine wird nicht aktiviert. Ansonsten sind die LEAVE-Funktionen der „REPOS“-Tabelle zu entnehmen: REPOS für OPEN≠REVERSE entspricht LEAVE für OPEN=REVERSE und umgekehrt.

Bei LEAVE für OPEN OUTPUT ist zu beachten, dass keine CLOSPOS-Routine aktiviert und die Bandposition nicht verändert wird.

Für Multifile-Bänder gilt: Wenn der Operand LEAVE nicht angegeben wird, wird beim CLOSE-Aufruf das Band an den Anfang zurückgespult.

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.

Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst. Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

REPOS
Für Banddateien:
positioniert ein Band abhängig von der LABEL-Angabe in FILE oder FCB auf den logischen Dateianfang.

Wurde im FILE-Aufruf der Operand BYPASS angegeben, so wird das Band zurückgespult, und die CLOSPOS-Routine wird nicht aktiviert. In allen anderen Fällen gilt die folgende Tabelle (Band positionieren):

LABEL-Angabe	**OPEN ≠ REVERSE**	**OPEN = REVERSE**
LABEL=(STD,n)	das Band wird automatisch auf HDR1 positioniert; FSEQ wird nicht verändert	das Band wird auf die Bandmarke hinter dem letzten EOF-Kennsatz der Datei positioniert; FSEQ wird um 1 erhöht
LABEL=NSTD	EXLST: CLOSPOS=NO, das Band wird automatisch auf die Bandanfangsmarke positioniert; FSEQ wird nicht verändert EXLST: CLOSPOS≠NO der Benutzer muss in einer Routine das Band selbst positionieren; FSEQ wird nicht verändert	EXLST: CLOSPOS=NO, das Band wird auf die Bandanfangsmarke positioniert; FSEQ=0 EXLST: CLOSPOS≠NO der Benutzer muss in der CLOSPOS-Routine das Band selbst positionieren; FSEQ wird nicht verändert
LABEL=NO		EXLST: CLOSPOS=NO, das Band wird automatisch auf die Abschnittsmarke hinter dem letzten Block der Datei positioniert, FSEQ um 1 erhöht

bei OPEN OUTPUT: keine CLOSPOS-Routine, keine Positionierung

RWD

Voreinstellung für Banddateien:

das Band wird zurückgespult und auf Bandanfang positioniert; FSEQ wird – auch bei Dateien mit NSTD-Kennsätzen – auf 0 gesetzt, d.h. FSEQ zeigt auf die erste Datei der Dateimenge/des Bandes.

Hinweis zur Programmierung

Der CLOSE-Makroaufruf zerstört die Register 0, 1, 14 und 15.

COMPFIL – Plattendateien vergleichen

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Der Makro COMPFIL vergleicht, analog zum Kommando COMPARE-DISK-FILES, zwei Plattendateien blockweise (UPAM) oder satzweise (SAM, ISAM) und informiert den Benutzer über das Vergleichsergebnis.

Es können auch temporäre Dateien oder Arbeitsdateien verglichen werden. Die Dateien können auf gemeinschaftlichen Datenträgern, Net-Storage oder Privatplatten liegen.

Die zu vergleichenden Dateien müssen in folgenden Merkmalen übereinstimmen:

- Zugriffsmethode (FILE-STRUCTURE bzw. FCBTYPE)
- Blockformat (BLOCK-CONTROL-INFO)
  Mit BLKCTRL=*IGNORE sind auch unterschiedliche Blockformate zulässig.
- Codierter Zeichensatz (CODED-CHARACTER-SET, EXTENDED_HOST_CODE)
- für SAM-Dateien:
  – RECORD-SIZE (RECSIZE) bei RECORD-FORM=F (RECFORM=F)
- für ISAM-Dateien:
  – RECORD-SIZE (RECSIZE) bei RECORD-FORM=F (RECFORM=F)
  – Struktur des ISAM-Schlüssels (KEY-LEN, KEY-POS, LOG-LEN und VAL-FL-LEN)
  – Struktur des Sekundärschlüssel (KEY-LEN, KEY-POS, DUPKEY) bei NK-ISAM
- für UPAM-Dateien:
  – BUFFER-LENGTH (BLKSIZE)
  – HIGHEST-USED-PAGE (LPP)

Dateien mit folgenden Eigenschaften können **nicht** verglichen werden:

- leere Dateien
- geöffnete Dateien
- gesperrte Dateien (z.B. SECURE-Lock)
- Kennzeichen REPAIR-NEEDED gesetzt
- Kennzeichen NO-DMS-ACCESS gesetzt

Dateigenerationsgruppen als Ganzes können nicht verglichen werden, wohl aber einzelne Dateigenerationen.

PLAM-Bibliotheken können blockweise verglichen werden. Ein Vergleich der enthaltenen Elemente ist nicht möglich.

Zum Vergleich von Banddateien steht das Dienstprogramm TPCOMP2 zur Verfügung, siehe Handbuch „Dienstprogramme“ [14].

Privilegierte Funktionen

Die Systembetreuung (Privileg TSOS) kann Dateien aller Benutzerkennungen vergleichen. Musterzeichen innerhalb der Benutzerkennung sind dabei nicht zulässig.

Format

Operation	Operanden
COMPFIL	,PATHNM1=<c-string 1..54: filename 1..54> / <var: char:54> ,PATHNM2=<c-string 1..54: filename 1..54> / <var: char:54> ,BLKCTRL=*IGNORE / *INCLUDE ,PAMINFO=*INCLUDE / *IGNORE ,OUTAREA=(NULL / <var: pointer>, 0 / <integer 0..32767> / <var: int:4>) ,CALLER=USER / SYSTEM ,EQUATES=YES / NO ,XPAND=PARAM / OUTPUT MF=L
	MF=D,PREFIX=D / <pre>
	MF=E,PARAM=<name 1..27>
	MF=C / M ,PREFIX=D / <pre> ,MACID=MAM / <macid>

Operandenbeschreibung

PATHNM1
Wählt die erste zu vergleichende Datei aus.

=<c-string 1..54: filename 1..54>
Name der ersten Datei.

=<var: char:54>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 54 Byte, in dem der Name der ersten Datei abgelegt ist.

PATHNM2
Wählt die zweite zu vergleichende Datei aus.

=<c-string 1..54: filename 1..54>
Name der zweiten Datei.

=<var: char:54>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 54 Byte, in dem der Name der zweiten Datei abgelegt ist.

BLKCTRL=*IGNORE / *INCLUDE
Gibt an, ob das Blockformat der Dateien in den Vergleich mit einbezogen wird (*INCLUDE) oder nicht (*IGNORE).

PAMINFO=*INCLUDE / *IGNORE
Gibt an, ob die Benutzerinformation im PAM-Schlüssel von UPAM-Dateien mit BLOCK-CONTROL-INFO=*PAMKEY in den Vergleich mit einbezogen wird (*INCLUDE) oder nicht (*IGNORE).

OUTAREA=(<address>,<length>)
Bestimmt Adresse und Länge des Ausgabebereichs, in dem die Informationen über den Dateivergleich abgelegt werden sollen.

<address>=NULL / <var: pointer>
Gibt die Adresse des Ausgabebereichs an.

<length>=0 / <integer 0..32767>
Gibt die Länge des Ausgabebereichs an.

<length>=<var: int:4>)
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 4 Byte, in dem die Länge des Ausgabebereichs abgelegt ist.

CALLER
Steuerungs-Operand nur für MF=E und MF=M:
Gibt an, ob beim Funktionsaufruf ein SVC oder ein direkter Aufruf generiert werden soll.

= USER
Der Funktionsaufruf wird über den SVC 144 generiert.

= SYSTEM
Steuerungs-Operand nur für Aufrufer aus TPR:
Es wird ein direkter Aufruf mit BASR generiert. Für die Schnittstelle gelten die DSL-Konventionen. Beim Binden von TU-Programmen kann der generierte Entry nicht befriedigt werden.

EQUATES
Steuerungs-Operand nur für MF=C und MF=D:
Gibt an, ob bei der Expansion des Parameter- oder Ausgabebereichs auch Equates für die Werte der Felder des Parameter- oder Ausgabebereichs generiert werden sollen.

= *YES
Bei der Expansion des Parameter- oder Ausgabebereichs werden auch Equates für die Werte der Felder des Parameter- oder Ausgabebereichs generiert.

= *NO
Bei der Expansion des Parameter- oder Ausgabebereichs werden keine Equates für die Werte der Felder des Parameter- oder Ausgabebereichs generiert.

XPAND
Steuerungs-Operand nur für MF=C und MF=D:
Es wird festgelegt, welche Struktur zu expandieren (erzeugen) ist. Angaben bei diesem Operanden werden bei anderen MF-Werten ignoriert.

= PARAM
Das Layout der Parameterliste wird expandiert.

= OUTPUT
Das Layout des Ausgabebereiches wird expandiert.

Hinweise zur Programmierung

1. Vor der Generierung des Layouts des Parameterbereichs muss der Standard-Header aufgelöst werden.
2. Alle RESERVED-Felder des Parameterbereichs müssen mit binären Nullen gelöscht sein.
3. Bei allen Änderungen im Parameterbereich, die nicht mit Hilfe von GCs vorgenommen werden, ist der Aufrufer selbst für die Konsistenz des Parameterbereichs verantwortlich.
4. Das Löschen des Ausgabebereichs obliegt dem Aufrufer.
5. Beim nichtprivilegierten Aufruf (Funktionszustand TU) zeigt Register 1 auf den Parameterbereich. Beim privilegierten Aufruf (Funktionszustand TPR) ist die Registerbelegung gemäß der DSL-Konvention.

Returncodes

Der Returncode wird im Standardheader des Parameterbereichs abgelegt. Der Parameterbereich darf dann nicht im Read-only-Bereich liegen, sonst erfolgt Programmterminierung.

Folgende Returncodes werden von COMPFIL erzeugt:

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'00'	X'0630'	Kein Fehler. Die Dateien sind gleich.
X'00'	X'00'	X'0631'	Kein Fehler. Die Dateien sind nicht gleich.
X'00'	X'40'	X'0501'	Dateikatalog nicht verfügbar
X'00'	X'40'	X'0505'	Fehler bei der Rechner-Kommunikation
X'00'	X'40'	X'0512'	Dateikatalog nicht gefunden
X'00'	X'40'	X'051B'	Benutzerkennung nicht im Pubset
X'00'	X'20'	X'0531'	Unerwarteter Fehler beim Dateikatalogzugriff
X'00'	X'01'	X'0554'	Format des Dateinamens unzulässig
X'00'	X'01'	X'0576'	Fehlerhafte Operandenkombination oder nicht gelöschte UNUSED-Felder
X'00'	X'82'	X'0594'	Nicht genug virtueller Speicher verfügbar
X'00'	X'20'	X'05AB'	Adresse des Ausgabebereichs falsch oder nicht angegeben
X'00'	X'20'	X'05C7'	Interner Fehler im DMS
X'00'	X'40'	X'05F4'	Angegebene Dateinamen sind gleich
X'00'	X'40'	X'05FC'	Benutzerkennung nicht im Home-Pubset
X'00'	X'01'	X'0624'	Ungültiger Dateiname
X'00'	X'40'	X'0636'	Dateiattribute sind nicht verträglich
X'02'	X'00'	X'06CB'	Ausgabeinformation nicht vollständig übertragen

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Layout des Parameterbereichs

Der Parameterbereich muss auf Wortgrenze ausgerichtet sein. Er beginnt mit einen Standardheader, den COMPFIL wie folgt initialisiert:

Function Unit Number	22
Function Number	33
Interface Version Number	1
Returncode	-1

Makroauflösung mit MF=D und XPAND=PARAM, sowie Standardwerten für EQUATES, PREFIX und MACID:

```
         COMPFIL MF=D,XPAND=PARAM
DMAVGLPL DSECT ,
DMAVHDR  DS    0A
DMAVFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
DMAVIFID DS    0A              0   INTERFACE IDENTIFIER
DMAVFCTU DS    AL2             0   FUNCTION UNIT NUMBER
DMAVFCT  DS    AL1             2   FUNCTION NUMBER
DMAVFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
DMAVRET  DS    0A              4   GENERAL RETURN CODE
DMAVSRET DS    0AL2            4   SUB RETURN CODE
DMAVSR2  DS    AL1             4   SUB RETURN CODE 2
DMAVSR1  DS    AL1             5   SUB RETURN CODE 1
DMAVMRET DS    0AL2            6   MAIN RETURN CODE
DMAVMR2  DS    AL1             6   MAIN RETURN CODE 2
DMAVMR1  DS    AL1             7   MAIN RETURN CODE 1
DMAVFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
DMAVPNAM1       DS    CL54              Pathname1
DMAVPNAM2       DS    CL54              Pathname2
DMAVBCTRL       DS    FL1               blockctrl operand
*   values of operand blckctrl_info
DMAVBLKIG       EQU   0                 ignore
DMAVBLKIN       EQU   1                 include
*
DMAVPINFO       DS    FL1               paminfo operand
*  values of operand paminfo
DMAVPIIG        EQU   0                 ignore
DMAVPIIN        EQU   1                 include
*
DMAVRES1        DS    XL6               RESERVED
DMAVARAD        DS    A                 Outarea=(<addr>,...)
DMAVARLN        DS    F                 Outarea=(...,<length>)
DMAV#           EQU   *-DMAVHDR
```

Layout des Ausgabebereichs

Der Ausgabebereich muss auf Wortgrenze ausgerichtet sein.

Makroauflösung mit MF=D und EXPAND=OUTPUT, sowie Standardwerten für EQUATES, PREFIX und MACID:

```
         COMPFIL MF=D,XPAND=OUTPUT
DMAVOUTP DSECT ,
*   Compfile Output
DMAVMSGNR      DS    F                   MESSAGENUMBER
*
DMAVMSGINF     DS    0XL80               MESSAGEINFO
DMAVMSG_DETAILS DS   0XL80               MESSAGEDETAILS
DMAVPNAM       DS    CL80                PATHNAME
           ORG   DMAVMSG_DETAILS
*
DMAVCMPINF     DS    0XL12               COMPAREINFO
DMAVPAGNR      DS    F                   PAGENUMBER
DMAVRECNR      DS    F                   RECORDNUMBER
DMAVBYTENR     DS    F                   BYTENUMBER
DMAVABSRECNR   DS    F                   ABSOLUT RECORDNUMBER FOR SAM
*                                        FILES
DMAVERRNR      DS    FL1                 ATTRIBUT ERROR
*   type of file attribut error
DMAVBKCTR      EQU   1                   blk-contr
DMAVFISTR      EQU   2                   file-struc
DMAVFITYP      EQU   3                   file-type
DMAVHIUSP      EQU   4                   high-us-pa
DMAVRECFR      EQU   5                   rec-form
DMAVRECSZ      EQU   6                   rec-size
DMAVKEYPO      EQU   7                   key-pos
DMAVKEYLN      EQU   8                   key-len
DMAVVALLN      EQU   9                   val-len
DMAVLOGLN      EQU   10                  log-len
DMAVALTIX      EQU   11                  alternate-index
DMAVLBP        EQU   12                  last-byte-pointer
DMAVBKSZ       EQU   13                  block-size
DMAVPKUI       EQU   14                  pamkey-user-info
DMAVLASTP      EQU   14                  last-position
*
         ORG   DMAVMSG_DETAILS+80
*
DMAVUNUS       DS    XL8                 UNUSED
DMAVOUTPUT#    EQU   *-DMAVMSGNR
```

Bei Ausgabe der COMPFIL-Information in den Ausgabebereich des Anwenders werden folgende Fälle unterschieden:

- Die Ausgabe war nicht möglich
  Vom Aufrufer wurde kein Ausgabebereich zur Verfügung gestellt bzw. der Ausgabebereich war schreibgeschützt. Der Anwender erhält im Standardheader des Parameterbereichs den Returncode X'05AB' nach Validierung des Ausgabebereiches bzw. der Adresse. War der Ausgabebereich zu klein um die Ausgabeinformation zu übertragen, dann erhält der Aufrufer den Returncode X'06CB'.
- Fehler beim Zugriff auf eine oder beide zu vergleichende Dateien
  Die erste Datei, bei der ein Zugriffsfehler auftritt, wird im Feld DMAVPNAM in der Form :<cat-id>:$<user-id>.<filename> ausgegeben. Außerdem erfolgt im Feld DMAVMSGNR die Ausgabe der Meldungsnummer X'0681' für einen allgemeinen Dateizugriffsfehler. Der genaue Grund für diesen Zugriffsfehler kann dem im Standardheader des Parameterbereichs abgelegten Returncode entnommen werden.
- Vergleich der beiden Dateien ist wegen der Unverträglichkeit ihrer Dateiattribute nicht möglich
  Im Standardheader des Parameterbereichs und im Feld DMAVMSGNR des Ausgabebereichs wird die Meldungsnummer X'0636' ausgegeben. Außerdem ist anhand des Feldes DMAVERRNR erkennen, welches Dateiattribut zum Abbruch des Vergleichs geführt hat.
- Für beide Dateien wurde der gleiche Namen angegeben
  Im Standardheader des Parameterbereichs und im Feld DMAVMSGNR des Ausgabebereichs wird der gleiche Returncode ausgegeben.
- Die beiden Dateien sind gleich
  Im Standardheader des Parameterbereiches wird der Returncode X'0630' ausgegeben. Das Feld DMAVMSGNR des Ausgabebereich enthält X'0000'.
- Die beiden Dateien sind nicht gleich
  Im Standardheader des Parameterbereiches wird der Returncode X'0631' ausgegeben. Abhängig von der Zugriffsmethode besteht folgender Zusammenhang zwischen MESSAGENUMBER im Feld DMAVMSGNR und der COMPARINFO in den Feldern DMAVPAGNR (PAGENUMBER), DMAVRECNR (RECORDNUMBER), DMAVBYTENR (BYTENUMBER) und DMAVABSRECNR (ABSOLUT RECORD NUMBER FOR SAM FILES).

Zugriffs-methode	MESSAGE-NUMBER	PAGE-NUMBER	RECORD-NUMBER	BYTE-NUMBER	ABSOLUT RECORD NUMBER	Bedeutung
SAM	X'0632'	<p>	<r>		<a>	[1]
ISAM	X'0633'		<r>			[2]
UPAM	X'0634'	<p>		<b>		[3]
UPAM	X'0635'	<p>				[4]

[1] Die beiden SAM-Dateien unterscheiden sich ab dem <a>-ten Datensatz. Das ist der <r>-te Datensatz innerhalb des <p>-ten 2K-Datenblocks.

[2] Die beiden ISAM-Dateien unterscheiden sich ab dem <r>-ten Datensatz

[3] Die beiden UPAM-Dateien unterscheiden sich ab dem <b>-ten Datenbyte innerhalb des <p>-ten 2K-Datenblocks

[4] Die beiden UPAM-Dateien unterscheiden sich in der Benutzerinformation im PAM-Schlüssel des <p>-ten 2K-Datenblocks, sind aber inhaltlich gleich (Angabe PAMINFO=*INCLUDE)

Beispiel für eine Aufruffolge

```
         MVC     COMPMFC(DMAV#),COMPMFL
         COMPFIL MF=M,OUTAREA=(A(COMPOAC),OUTLEN),PARAM=COMPMFC,        -
   ?                   PATHNM1=':X:SAM.1',PATHNM2=':X:SAM.2'
         COMPFIL MF=E,PARAM=COMPMFC
            .
            .
COMPMFC  COMPFIL MF=C,XPAND=PARAM
COMPOAC  COMPFIL MF=C,XPAND=OUTPUT
COMPMFL  COMPFIL MF=L,PATHNM1='AAA',PATHNM2='BBB'
OUTLEN   DC      A(DMAVOUTPUT#)
```

COPFILE – Datei kopieren

Makrotyp: S-Typ (C-Form/D-Form/E-Form/L-Form/M-Form) (siehe Seite 870)

Der COPFILE-Makroaufruf kopiert Dateien, Dateigenerationen und Dateigenerationsgruppen blockweise, ohne sie zu ändern. Er kann daher i. Allg. nicht verwendet werden, um Dateimerkmale zu verändern. Eine Ausnahme bildet lediglich die Möglichkeit, in bestimmten Fällen beim Kopieren die Blockkontrolleigenschaft der Datei zu ändern (siehe Abschnitt „Dateikettungsnamen“, Seite 220, und Beschreibung des Operanden BLKCTRL, Seite 223).

Wenn die Empfangsdatei (Kopie) noch nicht katalogisiert ist, dann wird sie bei Bearbeitung des COPFILE-Makroaufrufs automatisch auf gemeinschaftlichem Datenträger angelegt (wie bei einem FILE mit Standardwerten für die angegebene Empfangsdatei).

Soll die Empfangsdatei auf einem anderen Datenträger (Privatplatte, Net-Storage oder Band) gespeichert werden, muss sie vor Aufruf des COPFILE-Makros mit FILE (Operanden DEVICE, VOLUME) eingerichtet werden.

Ist die Plattendatei, in die kopiert werden soll, noch nicht katalogisiert, so wird die Primär- und Sekundärzuweisung von der Originalplattendatei übernommen.

Existiert die Zieldatei als Plattendatei, so werden deren Primär- und Sekundärzuweisung nur verändert, wenn sie kleiner sind als die der Originaldatei.

Liegt die Originaldatei auf Band, erhält die Zieldatei eine Standardzuweisung.

Hinweis

Der COPFILE-Makroaufruf ist eine Erweiterung des bisherigen COPY-Makroaufrufs um die Verwendung von Wildcards im Pfadnamen1 (Auswahl) und Pfadnamen2 (Konstruktion). Zusätzlich dazu werden die Operanden CHECK und LIST angeboten.
Die bisherige Funktionalität von COPY wird weiterhin unterstützt. Das Format des COPY-Makros wird deshalb im Anhang noch gezeigt (siehe Seite 901). Die Operanden des COPY-Makros entsprechen jedoch den Operanden des COPFILE-Makros. Sie sind deshalb nur hier beschrieben.

Dateigenerationsgruppen

Eine Dateigenerationsgruppe lässt sich nur dann in eine andere Dateigenerationsgruppe kopieren, wenn eine der folgenden Bedingungen erfüllt ist:

– Die Gruppeneinträge der beiden Dateigenerationsgruppen stimmen überein (d.h. die Werte von GEN, FIRSTGN, LASTGN und BASE sind gleich). Für die Dateigenerationsgruppe, in die das DVS die Kopie schreibt, müssen bereits die Generationen von FIRSTGN bis LASTGN katalogisiert sein.

– Der Wert von GEN ist für beide Dateigenerationsgruppen gleich und die Dateigenerationsgruppe, in die das DVS die Kopie schreibt, enthält noch keine Generation (d.h. FIRSTGN, LASTGN und BASE haben den Wert null).

Die zu kopierende Dateigenerationsgruppe darf keine Band-Dateigenerationen enthalten (COPFILE unterstützt kein Kopieren von Bändern).

Eine Dateigenerationsgruppe lässt sich nur dann in eine einzelne Datei oder Dateigeneration kopieren, wenn folgende Bedingungen erfüllt sind:

– Die Dateigenerationsgruppe besteht aus SAM-Dateigenerationen mit gleichen Eigenschaften (z.B. gleiche Satz- und Blocklänge, gleiches Satzformat, gleiche Blockkontrolleigenschaft).

– Die Dateigeneration, in die kopiert werden soll, gehört nicht der zu kopierenden Dateigenerationsgruppe an.

Eine einzelne Datei oder Dateigeneration lässt sich nur dann in eine Dateigenerationsgruppe kopieren, wenn folgende Bedingung erfüllt ist:

– Die Datei oder Dateigeneration muss den gleichen CODED-CHARACTER-SET besitzen wie die Dateigenerationsgruppe.

Dateien auf Privatplatten

Besitzt eine Datei auf privater Platte nur einen Eintrag im Systemkatalog, aber keinen im F1-Kennsatz, wird der Katalogeintrag gelöscht. Ist diese Datei die Eingabedatei, wird COPFILE zurückgewiesen.

Ein COPFILE-Aufruf für eine ISAM-Datei auf Privatplatten mit Index- und Datenteil auf verschiedenen Platten wird zurückgewiesen.

Banddateien

COPFILE arbeitet intern mit der Zugriffsmethode UPAM, die keine Folgebandverarbeitung zulässt. So können zwar mehrere Dateien auf ein Band kopiert werden (Makro FILE, Operand FSEQ). Es können jedoch keine Dateien kopiert werden, die sich über mehrere Bänder erstrecken.

– **K-Banddateien** (BLKCTRL=PAMKEY) müssen Standardblockformat haben (BLKSIZE=(STD,n)), damit der COPFILE-Makroaufruf sie bearbeiten kann.

– **NK-Banddateien** (BLKCTRL=DATA/NO) können von COPFILE verarbeitet werden, wenn ihr BLKSIZE-Wert ein Vielfaches von 2048 Byte beträgt.

  Werden NK-Dateien auf Band kopiert, erlischt die BLKCTRL-Information, wenn der Katalogeintrag gelöscht wird. Soll die Datei wieder zurückkopiert werden, muss dem COPFILE ein FILE-Aufruf mit den Operanden LINK und STATE=FOREIGN vorausgehen. Im FILE-Aufruf muss der Anwender den Operanden BLKCTRL richtig versorgen, d.h. dem tatsächlichen Dateiformat entsprechend NO oder DATA angeben.

Wird eine K-Datei (BLKCTRL=PAMKEY) auf diese Weise – versehentlich – als NK-Datei (BLKCTRL=DATA) kopiert, ist die entstehende Plattendatei nicht lesbar, da die ersten 16 Byte eines jeden logischen Blocks, die bei BLKCTRL=PAMKEY Daten enthalten, mit Verwaltungsinformationen überschrieben werden.

– **Fremddateien auf Band**: Soll eine Banddatei kopiert werden, die nicht katalogisiert ist, muss vor dem Kopieren ein TFT-Eintrag mit dem für COPFILE gültigen Dateikettungsnamen erstellt werden, um die Dateimerkmale festzulegen (siehe auch unter „Dateikettungsnamen“):
  ```
  FILE pfadname1,LINK=DMCOPY11,STATE=FOREIGN,BLKCTRL=...
  ```

Dateikettungsnamen

COPFILE arbeitet intern mit den Dateikettungsnamen DMCOPY11 (für die Originaldatei pfadname1) und DMCOPY22 (für die Zieldatei pfadname2). Nach Abschluss der Verarbeitung werden die Dateikettungsnamen implizit wieder freigegeben (impliziter REL-Makro).

Von der Möglichkeit, Original- und Zieldatei über einen FILE-Aufruf mit Dateikettungsnamen der COPFILE-Verarbeitung zuzuweisen, kann z.B. Gebrauch gemacht werden, um beim Kopieren die Blockkontrolleigenschaft der Datei zu verändern: Die Angabe des BLKCTRL-Operanden im FILE-Aufruf in Verbindung mit BLKCTRL=*IGNORE/*CHECK im COPFILE-Aufruf gestattet es nämlich, in bestimmten Fällen unterschiedliche BLKCTRL-Eigenschaften für Original- und Zieldatei beim Kopieren zu vereinbaren (siehe Beschreibung des Operanden BLKCTRL, Seite 223).

Im Zusammenhang mit Wildcards im Dateinamen wird ein existierender TFT-Eintrag (DMCOPY11/DMCOPY22) nur beim Kopieren der ersten zu bearbeitenden Datei wirksam.

Fern-Dateizugriff

(siehe auch Handbuch „RFA“ [6])

Das Kopieren von Fernsystem zu Fernsystem mit Eingabe und Ausgabe auf verschiedenen Systemen wird durch eine übergeordnete Ausführungsroutine unterstützt. Das lokale System dient hierbei nur als Zwischenstation beim Datentransfer. Vor dem Kopieren muss für beide Fernsysteme das Kommando SET-RFA-CONNECTION abgesetzt werden.

Wird eine Ferndatei mit dem Operanden PROTECT=*SAME auf eine lokale Datei kopiert, werden die Kennwörter nicht übernommen.

SM-Pubsets

Existiert die Zieldatei noch nicht, wird versucht, anhand der Eigenschaften der Quelldatei für die Volume-Set-Auswahl die Zieldatei auf einen geeigneten Volume-Set (Performance, Verfügbarkeit) anzulegen.

Dateiverschlüsselung

Zum Kopieren einer verschlüsselten Datei ist im Normalfall kein Crypto-Kennwort notwendig. COPFILE überträgt den Dateiinhalt einer verschlüsselten Datei, ohne die Datei zu entschlüsseln, und die Zieldatei erhält die gleichen Verschlüsselungsmerkmale wie die Ausgangsdatei, insbesondere das gleiche Crypto-Kennwort.

Ausnahmen sind Kopiervorgänge, die ein Entschlüsseln der Datei erfordern:
– Eine verschlüsselte Datei soll auf Band oder Privatplatte kopiert werden.
– Eine verschlüsselte Datei soll auf eine Dateigeneration kopiert werden.
– Über den TFT-Eintrag DMCOPY11 bzw. DMCOPY22 wurde Shared Update vereinbart.

Format

Operation	Operanden
COPFILE	BLKCTRL = *IGNORE / *CHECK / <var: blkctrl> ,CHDATE = *STD / *SAME / <var: bit: 1> ,CHECK = *MULTIPLE / *NO / *ERROR / *SINGLE / *CATALOG / *USERID /<var: check> ,*IGNORE* = *SOURCE / *TARGET / (*SOURCE,*TARGET) ,LIST = *NO / *SYSOUT / *ERRORS_TO_SYSOUT / <var: list> ,PATHNM1 = <c-string 1..80: filename 1..54 with-wild(80)> / <var: char: 80> ,PATHNM2 = <c-string 1..80: filename 1..54 with-constr-wild(80)>/ <var:char: 80> ,PROTECT = *STD / *SAME / *SAME-AND-CHANGE-DATE / <var: prot> ,REPLACE = *YES / *NO / <var: replace> ,MF = C / D / E / L / M ,PARAM = DMACOPPL / <adr> / <(r)> ,PREFIX = D / <pre> ,MACID = MAC / <macid>

Operandenbeschreibung

BLKCTRL
gibt an, ob sich die Zieldatei (bzw. der TFT-Eintrag DMCOPY22) hinsichtlich der BLKCTRL-Eigenschaft von der Quelldatei pfadname$_1$ unterscheiden darf.
TFT-Eintrag bzw. Zieldatei (bei Nulloperanden in der TFT) müssen die gleiche BLKCTRL-Eigenschaft besitzen wie die Quelldatei.

Voreinstellung: pfadname$_1$ und der TFT-Eintrag für DMCOPY22 müssen die gleiche BLKCTRL-Eigenschaft aufweisen.

= *IGNORE
Auch wenn die BLKCTRL-Attribute von pfadname$_1$ und der TFT-Eintrag für DMCOPY22 nicht übereinstimmen, ist in folgenden Fällen ein Kopieren von pfadname$_1$ auf pfadname$_2$ möglich:

BLKCTRL-Attribut der Datei pfadname$_1$	**BLKCTRL-Attribut der Datei pfadname$_2$**
PAMKEY	DATA (nur bei Plattendatei)
PAMKEY	NO
DATA (nur bei Plattendatei)	PAMKEY
NO	PAMKEY

Hinweis

Es liegt in der Verantwortung des Anwenders, Datenverluste beim Kopiervorgang auszuschließen. Solche Datenverluste können beim Kopieren einer Datei mit BLKCTRL= PAMKEY in eine Datei mit BLKCTRL=DATA oder NO auftreten: In beiden Fällen gehen die Informationen im Benutzerteil des PAM-Schlüssels verloren; hat die Zieldatei das Attribut BLKCTRL=DATA, werden zudem die ersten 12 Byte jedes logischen Blocks (bei ISAM-Dateien die ersten 16 Byte) mit dem Blockkontrollfeld überschrieben.

= *CHECK
Auch wenn die BLKCTRL-Attribute von pfadname$_1$ und der TFT-Eintrag für DMCOPY22 nicht übereinstimmen, ist in den Fällen ein Kopieren von pfadname$_1$ auf pfadname$_2$ möglich, in denen keine Anwenderinformationen im Benutzerteil des PAM-Schlüssels verloren gehen. Wenn der Benutzerteil des PAM-Schlüssels keine Anwenderinformationen enthält (dies wird hier geprüft), kann bei folgenden BLKCTRL-Attributen pfadname$_1$ auf pfadname$_2$ kopiert werden, andernfalls wird das Kommando zurückgewiesen.

BLKCTRL-Attribut der Datei pfadname$_1$	**BLKCTRL-Attribut der Datei pfadname$_2$**
PAMKEY	DATA (nur bei Plattendatei)
PAMKEY	NO

CHDATE
Gibt an, ob die Zieldatei das gleiche Änderungsdatum (CHANGE-DATE) erhält wie die Quelldatei.

= *STD
*Nur für PROTECT=*STD oder *SAME:*
Das Änderungsdatum der Zieldatei wird aktualisiert.

Die Angabe PROTECT=*SAME-AND-CHANGE-DATE wird noch kompatibel unterstützt und bewirkt, dass das Änderungsdatum der Quelldatei auf die Zieldatei übertragen wird.

= *SAME
Das Änderungsdatum der Quelldatei wird auf die Zieldatei übertragen. Die Angabe CHDATE=*SAME gilt auch in folgenden Fällen:
– Die Zieldatei liegt unter einer fremden Benutzerkennung.
– Die Zieldatei ist eine Dateigeneration.

CHECK
Nur für Wildcard-Angaben
Legt fest, unter welchen Bedingungen im Dialogbetrieb ein Anwenderdialog gestartet werden soll, wenn durch die Angabe von Wildcards mehrere Dateinamen selektiert werden.

Wird der Dialog gestartet, kann der Benutzer entscheiden, ob die Verarbeitung für die angezeigte(n) Datei(en) ausgeführt werden soll oder nicht. Er kann sich außerdem einen Hilfetext zu den Antwortmöglichkeiten ausgeben lassen sowie bei der Fortsetzung der Verarbeitung einen neuen Wert für CHECK und/oder LIST festlegen.

Im Stapelbetrieb gilt immer der Wert 'NO'.

Ohne Wildcards/Teilqualifikation in $pfadname_1$ ist der Operand wirkungslos.

= *MULTIPLE
Ein Kontrolldialog wird nur gestartet, wenn mehrere Dateien ausgewählt sind.
Sind Wildcards in der Katalog und/oder Benutzerkennung enthalten, dann wird für jeden Katalog und/oder jede Benutzerkennung ein Kontrolldialog durchgeführt.
Implizit gilt auch CHECK=*ERROR.

= *NO
Alle ausgewählten Dateien werden ohne Kontrolldialog, d.h. ohne Eingriffsmöglichkeit des Benutzers verarbeitet.

= *ERROR
Ein Fehler-Kontrolldialog wird gestartet, wenn bei der Verarbeitung eines ausgewählten Dateinamens ein Fehler auftritt. Ein Dateimenge-Kontrolldialog wird gestartet, wenn die Auswahlangabe mehr Dateien selektiert, als Speicherplatz für deren Verarbeitung verfügbar ist. Bei allen CHECK-Angaben $\neq$ *NO gilt implizit auch immer CHECK=*ERROR!

= *SINGLE
Für jeden ausgewählten Dateinamen wird ein Kontrolldialog durchgeführt. Implizit gilt auch CHECK=*ERROR.

= *CATALOG
Der Anwender muss in einem Kontrolldialog für jeden Katalog entscheiden, ob die darauf ausgewählten Dateien verarbeitet werden sollen.
Implizit gilt auch CHECK=*ERROR.

= *USERID
Nur für Systemverwalter:
Der Systemverwalter muss in einem Kontrolldialog für jede Benutzerkennung auf jedem Katalog entscheiden, ob die ausgewählten Dateien verarbeitet werden sollen.
Implizit gilt auch CHECK=*ERROR.

IGNORE
Nur für Systemverwalter:
Möglichkeit für den Systemverwalter, den Dateischutz für Quell- und/oder Zieldatei zu umgehen.
Die Angabe ist nicht wirksam für Dateien, die auf einem Fernrechner (RFA) liegen. Wenn für eine Datei unter fremder Benutzerkennung eine TSOS-Einschränkung vorliegt, wird das Schutzattribut ACCESS nicht ignoriert.

= *SOURCE
Die Schutzattribute READ-/EXEC-PASSWORD der Quelldatei werden beim Kopieren nicht beachtet. (Gilt auch für BASIC-ACL- bzw. GUARDS-Schutz.)

= *TARGET
Die Schutzattribute ACCESS/EXPIRATION-DATE sowie READ-/WRITE-/EXEC-PASSWORD der Zieldatei werden beim Kopieren nicht beachtet.
(Gilt auch für BASIC-ACL- bzw. GUARDS-Schutz.)

LIST
Bestimmt, ob ein Protokoll für alle mit Wildcards ausgewählten Dateinamen nach deren Abarbeitung nach SYSOUT geschrieben werden soll.

Ohne Wildcards/Teilqualifikation in $pfadname_1$ ist der Operand wirkungslos.

= *NO
Es soll nicht protokolliert werden.

= *SYSOUT
Jeder abgearbeitete Dateiname und eventuelle Fehler werden in einer Meldung protokolliert.

= *ERRORS_TO_SYSOUT
Nur Dateinamen, deren Abarbeitung zu Fehlern führte, werden in einer Meldung protokolliert.

MACID
wird nur in Verbindung mit MF=C ausgewertet und legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: MACID = MAC

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

= adr
adr ist die symbolische Adresse (der Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

PATHNM1
bezeichnet den Pfadnamen der Originaldatei.

= <c-string 1..80: filename 1..54 with-wild(80) without-gen>
Pfadname$_1$ (Angabe in Hochkommas)

= <var: char: 80: filename 1..54 with-wild(80) without-gen>
Name einer Variablen, die den Pfadnamen$_1$ enthält

Pfadname$_1$ bedeutet [:catid$_1$:][$userid$_1$.]dateiname$_1$

catid$_1$
Katalogkennung der Originaldatei;
Default-Catid: die der Benutzerkennung zugeordnete Katalogkennung

*$userid*$_1$
Benutzerkennung der Originaldatei;
Default-Userid: die Benutzerkennung des SET-LOGON-PARAMETERS- bzw. LOGON-Kommandos

dateiname$_1$
Name der Originaldatei, -dateigeneration oder -dateigenerationsgruppe

Für die Originaldatei muss Leserecht vorliegen.

Ist pfadname$_1$ eine Dateigenerationsgruppe, muss pfadname$_2$ ebenfalls eine Dateigenerationsgruppe sein.
(Ausnahme: die FGG pfadname$_1$ besteht aus SAM-Dateigenerationen mit gleichen Eigenschaften bzgl. Satzformat, Satz- und Blocklänge sowie Blockkontroll-Information. In diesem Fall kann in eine einzelne Datei oder in eine Dateigeneration kopiert werden. Diese Dateigeneration darf nicht zu der Dateigenerationsgruppe gehören, die kopiert werden soll.)

Wildcard-Angabe (Musterzeichen)
Auswahlangabe für die Dateien, die kopiert werden sollen. Der nichtprivilegierte Benutzer darf Musterzeichen nur in der Catid und im Dateinamen verwenden.

PATHNM2
bezeichnet den Pfadnamen der Ausgabe-/Zieldatei.

= <c-string 1..80: filename 1..54 with-wild(80) without-gen>
Pfadname$_2$ (Angabe in Hochkommas)

= <var: char: 80: filename 1..54 with-wild(80) without-gen>
Name einer Variablen, die den Pfadnamen$_2$ enthält

Pfadname$_2$ bedeutet [:catid$_2$:][$userid$_2$.]dateiname$_2$

catid$_2$
Katalogkennung der Ausgabedatei; Default-Catid: die der Benutzerkennung zugeordnete Katalogkennung

*$userid*$_2$
Benutzerkennung der Ausgabedatei; Default-Userid: die Benutzerkennung des SET-LOGON-PARAMETERS- bzw. LOGON-Kommandos

dateiname$_2$
Name der Ausgabedatei/-dateigeneration/-dateigenerationsgruppe

pfadname$_1$ und pfadname$_2$ dürfen nicht identisch sein.

Ist pfadname$_2$ noch nicht katalogisiert, darf nur die eigene Benutzerkennung angegeben werden, d.h. die des Kommandos SET-LOGON-PARAMETERS/LOGON oder eine Benutzerkennung, für die der Benutzer Mit-Eigentümer ist.

Ist pfadname$_2$ katalogisiert, muss **Schreibzugriff** möglich sein.

Der COPFILE-Makroaufruf wird zurückgewiesen, wenn pfadname$_2$ nur gelesen werden darf (z.B. ACCESS=READ oder EXDATE > Tagesdatum) oder wenn für die Plattendatei pfadname$_2$ die Sekundärzuweisung 0 ist und die Primärzuweisung nicht ausreicht.

Ist pfadname$_2$ unter einer fremden Benutzerkennung katalogisiert, muss außerdem die Benutzerkennung angegeben werden.

Ist pfadname$_2$ eine Dateigenerationsgruppe, muss auch pfadname$_1$ eine Dateigenerationsgruppe sein.

Wildcard-Angabe (Musterzeichen)
Konstruktionsangabe für die Dateien, in die auf Grund der Auswahlangabe (pfadname$_1$) kopiert werden sollen.

SM-Pubsets
Ist die Ausgabe-/Zieldatei noch nicht katalogisiert, wird anhand der Attribute der Originaldatei versucht, diese auf einem geeigneten Volume-Set anzulegen.

PREFIX
wird nur in Verbindung mit MF=C oder MF=D ausgewertet und legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein ein Zeichen langes Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen sollen.

PROTECT
Gibt an, ob die Kopie pfadname2 die gleichen Dateisicherungs- und Dateischutzmerkmale erhält wie pfadname1.
Die Verschlüsselungsmerkmale werden beim Kopieren in die Zieldatei übernommen, soweit möglich und erlaubt, unabhängig von den PROTECT-Angaben (siehe auch „Dateiverschlüsselung“ auf Seite 221).

= *STD
Ist pfadname$_2$ noch nicht katalogisiert, wird die neue Datei mit den Standardmerkmalen eingerichtet (vgl. Voreinstellungen der Operanden im CATAL-Makroaufruf, Seite 131, z.B. SHARE=NO, ACCESS=WRITE für Plattendateien usw.).

= *SAME
Die Kopie pfadname$_2$ erhält die gleichen Dateisicherungs- und Dateischutzmerkmale wie pfadname$_1$ (bzgl. ACCESS, BACKUP, DELDATE, DESTROY, LARGE, MANCLAS, MIGRATE (FORBIDDEN wird auf INHIBIT gesetzt), OPNBACK, RETPD, SHARE sowie die gleichen Kennwörter). Nicht übertragen werden: AUDIT, AVAIL, PREFORM, S0MIGR, STOCLAS, VOLSET und WORKFIL.

Die Angabe PROTECT=*SAME wird ignoriert, wenn $pfadname_2$ unter einer fremden Benutzerkennung katalogisiert oder eine Dateigeneration ist (Dateieigenschaften im Gruppeneintrag festgelegt!).

Wenn eine temporäre in eine permanente Datei kopiert wird, wird bei der Angabe von PROTECT=*SAME lediglich die Eigenschaft BACKUP=E übernommen. Die neue Datei wird bei ARCHIVE-Sicherungsläufen nicht berücksichtigt. Der BACKUP-Wert muss mit CATAL verändert werden, wenn die Datei mit ARCHIVE automatisch gesichert werden soll.

Wenn $pfadname_1$ durch einen Eintrag über BASIC-ACL (siehe Handbuch „Einführung in das DVS“ [1]) oder GUARDS geschützt ist, so gilt beim Kopieren mit PROTECT= *SAME:

– Wird die Zieldatei auf einer gemeinschaftlichen Platte angelegt, so werden die Zugriffsrechte von BASIC-ACL oder GUARDS kopiert.

  Wird die Zieldatei $pfadname_2$ auf einer privaten Platte angelegt und ist $pfadname_1$ über BASIC-ACL geschützt, so werden für $pfadname_2$ die Schutzmerkmale der BASIC-ACL übernommen. Ist für $pfadname_1$ ein GUARDS-Eintrag angelegt, so wird $pfadname_2$ mit den Standard-Schutzattributen SHARE=NO und ACCESS= WRITE versehen.

– Wird die Zieldatei $pfadname_2$ auf einem Magnetband angelegt, so wird sie mit den Standard-Schutzattributen SHARE=YES und ACCESS=WRITE versehen, unabhängig von den Schutzmerkmalen, die über BASIC-ACL oder GUARDS für $pfadname_1$ vereinbart wurden.

– Ist die Quelldatei $pfadname_1$ nicht unter der Benutzerkennung katalogisiert, in der COPFILE aufgerufen wird, so wird $pfadname_2$ – unabhängig von den Schutzmerkmalen, die über BASIC-ACL oder GUARDS für $pfadname_1$ vereinbart wurden – mit Standard-Schutzattributen versehen; das sind USER-ACCESS=OWNER-ONLY und ACCESS=WRITE bei einer Plattendatei, USER-ACCESS=ALL-USERS und ACCESS=WRITE bei einer Banddatei.

Beim Kopieren auf Band kann die Schutzfrist (EXDATE) nur Werte bis zu einer Differenz von 32767 aufnehmen (für größere Werte wird der Maximalwert angenommen).

= *SAME-AND-CHANGE-DATE
Die Angabe wirkt wie PROTECT=*SAME. Zusätzlich wird das Änderungsdatum (CHANGE-DATE) der Quelldatei auf die Zieldatei übertragen.

Die Angabe PROTECT=*SAME-AND-CHANGE-DATE wird nur noch kompatibel unterstützt. Um das Änderungsdatum der Quelldatei zu übertragen sollte der Operand CHDATE=*SAME verwendet werden.

REPLACE
Der Anwender kann steuern, ob eine bereits vorhandene Ausgabedatei $pfadname_2$ überschrieben wird.
Ist $pfadname_2$ eine Banddatei oder leer, wird der Operand ignoriert, die „alte“ Datei wird ohne Meldung überschrieben.

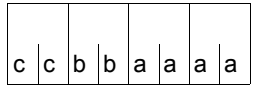
= *YES
$pfadname_2$ wird ohne Meldung überschrieben.

= *NO
$pfadname_2$ wird nicht überschrieben. Der Aufruf wird mit dem Fehlercode X'051A' abgewiesen.

Returncodes

Der Fehlercode wird nur noch im Standardheader und nicht mehr wie beim COPY-Makroaufruf im Mehrzweckregister 15 zurückgeliefert. Ist der Parameterbereich nicht zugreifbar oder kürzer als die Länge des Standard-Headers, oder trat ein Ausrichtungsfehler auf, dann wird eine Programm-Terminierung mit STXIT-Anschluss eingeleitet. Die Fehlercodes sind in den Makros DMAIDEM/DCOIDEM beschrieben.

Standard-header:



Über die Ausführung des Makros COPFILE wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Kein Fehler
X'01'	X'00'	X'0000'	Nur im Zusammenhang mit Kontrolldialogen: Der Auftrag wurde im Dialog ganz oder teilweise zurückgezogen, d.h. mindestens ein Kontrolldialog wurde mit *NO beantwortet
X'02'	X'00'	X'0000'	Nur im Zusammenhang mit CHECK≠NO : Es ist zwar ein Fehler aufgetreten, aber in einem Fehlerdialog wurde die Fortsetzung der Funktion gefordert
	X'40'	X'0501'	Angeforderter Katalog nicht verfügbar
	X'82'	X'0502'	Angeforderter Katalog im Ruhezustand
	X'40'	X'0503'	Falsche Information im MRSCAT
	X'82'	X'0504'	Fehler im Katalogverwaltungssystem
	X'40'	X'0505'	Fehler bei der Rechner-Kommunikation (MRS)
	X'80'	X'0506'	Operation wegen Masterwechsel abgebrochen
	X'40'	X'0510'	Fehler beim Aufruf einer internen Funktion
	X'40'	X'0512'	Angeforderter Katalog unbekannt

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'051A'	Datei existiert bereits
	X'40'	X'051B'	Benutzerkennung im angegebenen Pubset unbekannt
	X'40'	X'051C'	Kein Zugriffsrecht auf angegebenen Pubset
	X'40'	X'051D'	LOGON-Passwort auf angegebenem Pubset anders
	X'20'	X'0530'	Fehler bei Speicherplatz-Anforderung
	X'20'	X'0531'	Unerwarteter Fehler beim Katalogzugriff
	X'40'	X'0533'	Angegebene Datei nicht gefunden
	X'82'	X'0534'	Privater Datenträger kann nicht zugewiesen werden
	X'40'	X'0535'	Keine Zugriffsberechtigung auf den Katalogeintrag der Datei (nur im Zusammenhang mit der CCS-Vergabe auf fremder Benutzerkennung)
	X'20'	X'053B'	Systemfehler beim Dateizugriff
	X'82'	X'053C'	Katalog-Datei des Pubsets ist voll
	X'40'	X'053D'	Katalog oder F1-Etikett-Block ist zerstört
	X'40'	X'053E'	Datei auf privatem Datenträger bereits katalogisiert
	X'82'	X'053F'	Datei ist von einer anderen Task reserviert
	X'01'	X'0576'	Widersprüchliche Operandenkombination oder reservierte Felder des Parameterbereiches verwendet
	X'20'	X'0577'	Interner Fehler beim Zugriff auf die Auftragsumgebung
	X'82'	X'0594'	Nicht genug virtueller Speicher verfügbar. Dieser Returncode kann insbesondere auch im Zusammenhang mit einer Auswahlangabe (Wildcard) auftreten, wenn zu viele Dateien selektiert werden
	X'01'	X'0599'	Operand wird in der RFA-BS-Version nicht unterstützt
	X'01'	X'05A7'	Erster Dateiname fehlerhaft
	X'01'	X'05A9'	Zweiter Dateiname fehlerhaft
	X'20'	X'05C7'	Interner Fehler im DVS
	X'01'	X'05EE'	Dateiname zu lang
	X'01'	X'05F0'	Fremde Benutzerkennung für Datei2 nicht erlaubt
	X'01'	X'05F1'	Kopieren in die angegebene Datei nicht möglich
	X'01'	X'05F2'	Unzulässige Angabe von *DUMMY
	X'40'	X'05F3'	Erste oder zweite Datei geschützt
	X'01'	X'05F4'	Erster und zweiter Dateiname sind gleich
	X'20'	X'05F5'	Einige Blöcke konnten nicht kopiert werden
	X'01'	X'05F6'	Datei nicht kopierbar
	X'40'	X'05F9'	Unvereinbare Eigenschaften von Quell- und Zieldatei

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'05FC'	Angegebene Benutzerkennung nicht im Home-Pubset
	X'40'	X'0610'	Mindestens für einen der ausgewählten Dateinamen lieferte die Funktionsausführung einen Returncode
	X'01'	X'0611'	Fehlerhafte Konstruktionsangabe (Operand PATHNM2 im Zusammenhang mit Wildcards)
	X'40'	X'0666'	Die Datei ist durch ACL oder GUARDS schreibgeschützt
	X'40'	X'0698'	Dateigenerationsgruppen haben nicht die gleichen Attribute
	X'40'	X'06B5'	Datei ist nicht ordnungsgemäß geschlossen
	X'40'	X'06B6'	Attribute der Datei passen nicht zur Dateigenerationsgruppe
	X'40'	X'06C4'	Dateigenerationsgruppe noch nicht katalogisiert
	X'01'	X'06C7'	Ungültige Generationsnummer angegeben
	X'40'	X'06CC'	nur bei Auswahlangabe (Wildcard): keine Datei entspricht der Auswahlangabe
	X'01'	X'06D7'	Generationsgruppe kann nicht in einzelne Generation dieser Gruppe kopiert werden
	X'01'	X'06D8'	Generationen der angegebenen Gruppe haben unterschiedliche Dateimerkmale
	X'01'	X'06DE'	Datei oder Generation kann nicht in eine Gruppe kopiert werden
	X'01'	X'06FD'	Parameterbereich ungültig oder nicht zugreifbar
	X'40'	X'06FF'	BCAM-Verbindung abgebrochen
	X'01'	X'FFFF'	Falsche Funktionsnummer im Parameterbereichs-Header
	X'03'	X'FFFF'	Falsche Versionsnummer im Parameterbereichs-Header

CREAIX – Sekundärschlüssel für ISAM-Datei erzeugen

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form) (siehe Seite 870)

Der Makro CREAIX definiert einen oder mehrere Sekundärschlüssel (maximal 30) für eine NK-ISAM-Datei. Insgesamt können für eine Datei bis zu 30 Sekundärschlüssel vereinbart werden; sie sind jeweils durch ihren im CREAIX-Aufruf festgelegten Namen zu identifizieren. Über seinen Namen kann jeder dieser Sekundärschlüssel in den Makros GET, GETR, GETKY sowie SETL angesprochen werden und ermöglicht es so dem Anwender, auf Datensätze anhand von Sekundärschlüsselwerten zuzugreifen.

Zur Einrichtung von Sekundärschlüsseln werden zunächst alle Sätze der Datei sequenziell gelesen. Zu jedem Satz wird ein Tripel – bestehend aus dem Index in der Liste der zu erstellenden Sekundärschlüssel, dem jeweiligen Sekundärschlüssel selbst und dem jeweiligen Primärschlüssel – gebildet. Diese Einheiten werden anschließend pro Sekundärschlüssel nach den Sekundärschlüsselwerten sortiert und, mit Zeitstempeln versehen, als Einträge in die Sekundärindexblöcke übernommen, die für diesen Sekundärschlüssel angelegt werden.

Sekundärschlüssel können nur für existierende NK-ISAM-Dateien eingerichtet werden, d.h. für Dateien, die schon einmal im Modus OUTPUT oder oder OUTIN eröffnet waren. Darüberhinaus dürfen in einer NK-ISAM-Datei, für die Sekundärschlüssel vereinbart werden sollen, Primärschlüsselwerte nicht mehrfach vorkommen (keine DUPEKYs!), und es dürfen weder Logische noch Wertmarkierungen definiert sein. Zum Zeitpunkt des Makroaufrufs darf für die NK-ISAM-Datei nicht (durch ein ADD-FILE-LINK-Kommando) SHARED-UPDATE=*YES bzw. (durch ein Makro) SHARUPD=YES eingestellt sein, und während der Erstellung des Sekundärschlüssels können keine anderen Benutzer auf sie zugreifen.

Wird während der Erstellung des Sekundärschlüssels das Programm abgebrochen, so wird der Sekundärschlüssel im Kontrollblock dieser Datei als unvollständig markiert. Beim Versuch, eine solche Datei zu öffnen wird – falls im Programm vorgesehen – der OPENER-Ausgang genommen und im FCB der Fehlerschlüssel 0D84 abgelegt. Erst nach dem Löschen des unvollständigen Sekundärschlüssels (und ggf. seiner erneuten Vereinbarung mit CREAIX) kann die Datei wieder geöffnet werden.

Aus Performancegründen empfiehlt es sich, Sekundärschlüssel für eine Datei erst zu definieren, wenn sie bereits mit Datensätzen geladen ist.

Format

Operation	Operanden
CREAIX	[,DUPKEY=({YES / NO}[,{YES / NO},...])] ,KEYLEN=(keylen1 [,keylen2,...]) ,KEYNAME=(keyname1 [,keyname2,...]) ,KEYPOS=(keypos1 [,keypos2,...]) ,{LINK=linkname1 / FILE=pfadname} [,SORTLNK=linkname2] [,VERSION={1 / 2}] MF=L
	MF=E,PARAM={adr / (r)}
	MF=D[,PREFIX={D / pre}] [,VERSION={1 / 2}]
	MF=C[,PREFIX={D / pre}][,MACID={ISS / macid}] [,VERSION={1 / 2}]

Operandenbeschreibung

DUPKEY
legt für jeden anzulegenden Sekundärschlüssel (Operand KEYNAME) fest, ob gleiche Werte in verschiedenen Datensätzen vorkommen dürfen.

Die runden Klammern in dieser Angabe können weggelassen werden, wenn die Liste nur eine Spezifikation für DUPKEY enthält. Die Angabe einer Liste ist nur für VERSION=2 zulässig, für VERSION=1 ist nur eine Angabe ohne Klammern erlaubt.

= YES
ist Voreinstellung: Gleiche Werte des Sekundärschlüssels dürfen in der Datei mehrfach auftreten.

= NO
In der Datei dürfen verschiedene Datensätze nicht ein und denselben Wert für den Sekundärschlüssel enthalten.

FILE = pfadname
Bezeichnet die NK-ISAM-Datei, für die der Sekundärschlüssel vereinbart werden soll, mit: <c-string 1..54: filename 1..54>

Die Datei muss zum Zeitpunkt des CREAIX-Aufrufs mindestens einmal im Modus OUTPUT oder OUTIN eröffnet worden sein und darf keine mehrfach auftretenden Primärschlüsselwerte, keine logischen Markierungen und keine Wertmarkierungen enthalten. Die Angabe im FILE-Operanden wird ignoriert, wenn zugleich der LINK-Operand angegeben ist.

Pfadname bedeutet [:catid:][$userid.]dateiname

catid
Katalogkennung: falls nicht angegeben, wird die Default-Catid der Benutzerkennung angenommen.

userid
Benutzerkennung: falls nicht angegeben, wird die Benutzerkennung des SET-LOGON-PARAMETERS- bzw. des LOGON-Kommandos angenommen.

dateiname
vollqualifizierter Dateiname

KEYLEN = (keylen1 [,keylen2,...])
vereinbart für jeden anzulegenden Sekundärschlüssel (Operand KEYNAME) die Länge (in Byte). „keylen“ ist eine beliebige ganze Zahl mit $1 \leq$ keylen ≤ 127.
KEYPOS und KEYLEN sind so zu wählen, dass der Sekundärschlüssel
– auch im kürzesten Satz der Datei noch vollständig enthalten ist und
– ganz in einem Datenblock liegt und nicht in einen Überlaufblock hineinragt.

Die runden Klammern in dieser Angabe können weggelassen werden, wenn die Liste nur eine Spezifikation für KEYLEN enthält. Die Angabe einer Liste ist nur für VERSION=2 zulässig, für VERSION=1 ist nur eine Angabe ohne Klammern erlaubt.

KEYNAME = (keyname1 [,keyname2,...])
legt den Namen der anzulegenden Sekundärschlüssel fest. In der Liste dürfen maximal 30 Namen angegeben werden, wobei zu beachten ist. dass für eine NK-ISAM-Datei maximal 30 Sekundärschlüssel insgesamt erzeugt werden können.

Der Name darf nicht bereits für einen anderen Sekundärschlüssel vereinbart sein. „keyname“ darf bis zu acht Zeichen lang sein und alle Buchstaben und Ziffern sowie die Sonderzeichen „$“, „#“ und „@“ enthalten. „keyname“ muss mit einem Buchstaben oder einem Sonderzeichen beginnen.

Die runden Klammern in dieser Angabe können weggelassen werden, wenn die Liste nur einen Namen enthält. Die Angabe einer Liste ist nur für VERSION=2 zulässig, für VERSION=1 ist nur eine Angabe ohne Klammern erlaubt.

KEYPOS = (keypos1 [,keypos2,...])
gibt für jeden anzulegenden Sekundärschlüssel (Operand KEYNAME) die Position des ersten Zeichens im Datensatz an.
Für „keypos“ ist eine beliebige ganze Zahl mit $1 \leq$ keypos ≤ 32496 erlaubt. Bei Sätzen variabler Länge müssen 4 Byte für das Satzlängen- und Steuerfeld berücksichtigt werden.
KEYPOS und KEYLEN sind so zu wählen, dass der Sekundärschlüssel
– auch im kürzesten Satz der Datei noch vollständig enthalten ist und
– ganz in einem Datenblock liegt und nicht in einen Überlaufblock hineinragt.

Die runden Klammern in dieser Angabe können weggelassen werden, wenn die Liste nur eine Spezifikation für KEYPOS enthält. Die Angabe einer Liste ist nur für VERSION=2 zulässig, für VERSION=1 ist nur eine Angabe ohne Klammern erlaubt.

LINK = linkname1
Legt den Dateikettungsnamen der Datei fest, für die der Sekundärschlüssel vereinbart werden soll. Zum Programmablauf muss diesem Dateikettungsnamen eine NK-ISAM-Datei zugeordnet werden. Sie muss zum Zeitpunkt des CREAIX-Aufrufs mindestens einmal im Modus OUTPUT oder OUTIN eröffnet worden sein und darf keine mehrfach auftretenden Primärschlüsselwerte, keine logischen Markierungen und keine Wertmarkierungen enthalten. „linkname1“ darf bis zu acht Zeichen lang sein. Soll der Dateikettungsname über die Kommandoschnittstelle ansprechbar sein, muss er dem Datentyp <structured_name 1..8> entsprechen (siehe Handbuch „Kommandos“ [3]).

MACID
legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: MACID = ISS

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

= adr
adr ist die symbolische Adresse (der Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

PREFIX
legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein Zeichen langes Präfix, mit dem die generierten Feldnamen und Equates beginnen sollen.

SORTLNK = linkname2
Legt den Dateikettungsnamen einer Arbeitsdatei für das Sortierprogramm fest. Diese Arbeitsdatei wird nur benutzt, wenn der virtuelle Adressraum nicht zum Sortieren der Einträge für den Sekundärindexblock ausreicht.
Wenn eine Arbeitsdatei zum Sortieren benötigt wird und SORTLNK nicht angegeben oder dem Dateikettungsnamen beim Programmablauf keine Datei zugeordnet ist, so legt der Makro eine Arbeitsdatei mit dem Namen DISWORK.tsn an (tsn: TSN des Auftrags, der den Makro aufruft).
„linkname2“ darf bis zu acht Zeichen lang sein, und muss nach den Regeln für Dateinamen aus Buchstaben, Ziffern und Sonderzeichen gebildet werden.

VERSION
gibt die Version des generierten Makros CREAIX an.

= 1
Es wird die „alte“ Makroversion generiert.

= 2
Es wird die ab BS2000/OSD-BC V3.0 gültige Version des Makros CREAIX generiert.

Hinweise zur Programmierung

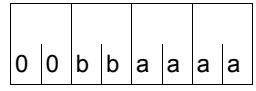
1. Die D-Form und C-Form des Makros generieren Feldnamen und EQU-Anweisungen für Returncodes. Sie beginnen mit der Zeichenfolge DISS..., die durch die Operanden PREFIX und MACID modifiziert werden kann.
2. Ist bei der D-Form keine symbolische Adresse angegeben, wird der DSECT-Name DISCRAIX generiert, wobei das erste Zeichen durch eine PREFIX-Angabe modifiziert wird.
3. Bei der Expansion des Makros CREAIX wird in der Parameterliste ein Feld mit dem Namen KEY# (mit Standard-Präfix DISS oder entsprechend verändert durch die Operanden PREFIX und MACID) erzeugt. Dieses Feld enthält die Anzahl der anzulegenden Sekundärindizes (maximal 30), wobei die Versorgung durch die Expansion des Makros erfolgt. Wenn die Parameterliste allerdings erst zur Programm-Laufzeit dynamisch aufgebaut wird, muss das Feld KEY# vom Programm explizit versorgt werden.
4. Beim Auftreten eines Fehlers enthält die Parameterliste den Index des Sekundärindex in der angegebenen Liste, bei dem dieser Fehler auftrat. Außerdem wird ein eventuell aufgetretener DMS-Fehler ebenfalls in der Parameterliste hinterlegt (der Name des Feldes, das den Index des Sekundärindex beinhaltet, bei dem ein Fehler auftrat, lautet DISAERR bzw. <xxxy>AERR je nach PREFIX und MACID).
5. Für eine NK-ISAM-Datei können bis zu maximal 30 Sekundärindizes insgesamt erzeugt werden. Es ist also darauf zu achten, dass einerseits die im Makro spezifizierte Liste der Namen der zu definierenden Sekundärindizes höchstens 30 Elemente umfasst. Andererseits darf die Summe aus bereits existierenden und neu zu erzeugenden Sekundärindizes ebenfalls nicht größer als 30 sein (beide Fälle führen zu einem entsprechenden Returncode).
6. Es ist darauf zu achten, dass für eine Parameterliste der Wert von VERSION konsistent für Aufrufe mit unterschiedlichen MF-Formaten versorgt wird.

Returncodes

Die vom Makro CREAIX ausgegebenen Returncodes werden im Standardheader der Operandenliste hinterlegt. Der Standardheader muss vor der Generierung der DSECT für die CREAIX-Parameterliste definiert werden.

Die mit der C- oder D-Form des Makros generierten Returncodes beginnen standardmäßig mit der Zeichenfolge DISS, diese kann im ersten Zeichen durch die Angabe von PREFIX, in den Zeichen 2 bis 4 durch die Angabe von MACID entsprechend modifiziert werden.

Standard-header:



Über die Ausführung des Makros CREAIX wird im Standardheader folgender Returncode übergeben (bb = SUBCODE1, aaaa = MAINCODE):

X'bb'	**X'aaaa'**	**Erläuterung**
X'00'	X'0000'	Funktion erfolgreich ausgeführt.
X'01'	X'0001'	Die Operandenliste ist nicht verfügbar.
X'40'	X'0002'	Sekundärschlüssel werden im fernen System nicht unterstützt (bei Makroaufruf über RFA).
X'40'	X'0003'	Die angegebene Katalogkennung existiert nicht.
X'40'	X'0004'	Auf den Katalog kann nicht zugegriffen werden.
X'01'	X'0005'	Die Operandenliste enthält einen ungültigen Namen.
X'40'	X'0006'	Die angegebene Datei enthält DUPEKYs.
X'40'	X'0007'	Der zu definierende Sekundärschlüssel existiert bereits.
X'01'	X'0009'	ungültige Angabe für KEYLEN.
X'40'	X'000A'	Die angegebene Datei enthält Logische oder Wertmarkierungen
X'20'	X'000B'	Systemfehler.
X'40'	X'000C'	Der Benutzeradressraum ist zu klein.
X'01'	X'000D'	ungültige Angabe für KEYPOS.
X'40'	X'000E'	Der Kontrollblock der Datei ist fehlerhaft.
X'40'	X'000F'	Ein Satz in der angegebenen Datei ist zu klein für den zu definierenden Sekundärschlüssel.
X'40'	X'0010'	Es sind bereits 30 Sekundärschlüssel für die Datei definiert.
X'40'	X'0011'	Die Datei enthält unvollständige Sekundärindexblöcke.
X'40'	X'0012'	Der ISAM-Pool ist überlastet.
X'40'	X'0013'	Der Sekundärschlüssel ist bereits mit anderen Attributen definiert.
X'40'	X'0014'	Unterbrechung durch CANCEL
X'40'	X'0015'	Unterbrechung durch BREAK
X'01'	X'0017'	In der Operandenliste wurde keine Datei spezifiziert.

X'bb'	X'aaaa'	Erläuterung
X'40'	X'0018'	Zum Zeitpunkt des Makroaufrufs ist SHARUPD=YES eingestellt.
X'40'	X'0019'	Der Dateikettungsname ist fehlerhaft.
X'40'	X'001A'	Obwohl DUPKEY=NO gefordert ist, kommen gleiche Werte des Sekundärschlüssels in verschiedenen Sätzen vor.
X'40'	X'001B'	ungültiges Listenelement
X'40'	X'001C'	ungültige Anzahl von Sekundärindizes in der Liste
X'40'	X'0040'	OPEN-Fehler.
X'40'	X'0041'	CLOSE-Fehler.
X'40'	X'0042'	Beim Schreiben der Sekundärindexblöcke trat ein Fehler auf.
X'40'	X'0043'	Beim Lesen der Datei trat ein Fehler auf.
X'40'	X'0044'	Die Datei ist keine NK-ISAM-Datei.
X'40'	X'0081'	Beim Sortieren der Sekundärindexeinträge trat ein DVS-Sonderzustand ein.
X'40'	X'0082'	Beim Sortieren der Sekundärindexeinträge trat ein interner Fehler auf.
X'01'	X'FFFF'	Linkage-Fehler (Funktion nicht unterstützt)
X'02'	X'FFFF'	Linkage-Fehler (Funktion nicht verfügbar)
X'03'	X'FFFF'	Linkage-Fehler (Version nicht unterstützt)

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

CREPOOL – ISAM-Pool erzeugen

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form) (siehe Seite 870)

Der Makro CREPOOL erzeugt einen task- oder host- oder userid-spezifischen ISAM-Pool oder verbindet den Auftrag mit einem bereits existierenden ISAM-Pool. Der ISAM-Pool ist eindeutig gekennzeichnet durch folgende Eigenschaften:

– Poolnamen: Operand NAME
– Katalogkennung: Operand CATID
– Geltungsbereich: Operand SCOPE
– Größe des ISAM-Pools: Operand SIZE
– Art der Pufferung: Operand WROUT
– Performance-Eigenschaft des ISAM-Pools: Operand RESDNT

Der Makro CREPOOL kann nur bei XS-Programmierung (31-Bit-Schnittstelle) genutzt werden.

Hinweis

Taskübergreifende ISAM-Pools (SCOPE=HOST) werden bei der Dateieröffnung automatisch dateispezifisch in einem Data Space angelegt.
Die bis BS2000/OSD-BC V6.0A vorhandenen Geltungsbereiche SCOPE=USERID und SCOPE=USERGROUP werden aus Kompatibilitätsgründen noch akzeptiert, intern jedoch auf SCOPE=HOST (taskübergreifender ISAM-Pool) abgebildet. Für weitere Informationen zu ISAM-Pools in Data Spaces siehe Handbuch „Einführung in das DVS“ [1].

Format

Operation	Operanden
CREPOOL	[,CATID=catid] [,MODE={ANY | NEW}] NAME=poolname [,SCOPE={TASK | USERID | USERGROUP | HOST}] [,SIZE={STD | zahl}] [,RESDNT={NO | YES}] [,WROUT={YES | NO | UNCOND-NO}] MF=L
	MF=E,PARAM={adr | (r)}
	MF=D[,PREFIX=pre]
	MF=C[,PREFIX=pre][,MACID=macid]

Operandenbeschreibung

CATID = catid
gibt die Katalogkennung des Pubsets an, dem der ISAM-Pool zugeordnet werden soll: der ISAM-Pool wird an dem Host-Rechner eingerichtet, zu dem der hier genannte Pubset gehört. Die Katalogkennung kann wie beim Dateinamen als Teil des Namens gesehen werden, d.h. verschiedene Katalogkennungen bezeichnen verschiedene ISAM-Pools.

Voreinstellung: Default-Catid des aufrufenden Auftrags

MACID
wird nur in Verbindung mit MF=C ausgewertet und legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: MACID = ISC

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

MODE
Gibt für taskübergreifende ISAM-Pools an, ob ein neuer ISAM-Pool angelegt werden soll oder ob eine Verbindung zu einem bereits existierenden Pool gleichen Namens und gleicher Catid hergestellt werden kann.

Standardmäßig schließt das DVS den Auftrag unbedingt an einen bereits bestehenden ISAM-Pool mit o.g. Namen an.

= ANY
Falls bereits ein taskübergreifender ISAM-Pool mit gleichem Namen und gleicher Catid von einer anderen Task erzeugt wurde, wird der Auftrag an diesen Pool angeschlossen, auch wenn die Angabe im Operanden SIZE nicht mit der tatsächlichen Poolgröße übereinstimmt.
Existiert noch kein solcher ISAM-Pool, wird ein neuer Pool mit der in SIZE definierten Größe angelegt.

Mit der Spezifikation des Parameters CREATION-MODE=ANY kann **ausschließlich** die Verbindung zu einem von einer anderen Task erzeugten ISAM-Pool hergestellt werden, d.h. wenn ein Task zweimal (oder mehrmals) hintereinander ein CREPOOL mit CREATION-MODE=ANY für **einen** ISAM-Pool abgesetzt wird, so wird der 2. Aufruf mit

Fehler abgewiesen, auch wenn der ISAM-Pool durch den 1. Aufruf bereits existiert. D.h. wiederum: Innerhalb einer Task kann **ein** ISAM-Pool nur kreiert werden, wenn er für diese Task noch nicht existiert.

= NEW
Es soll ein taskübergreifender ISAM-Pool angelegt werden; existiert jedoch bereits ein solcher Pool gleichen Namens und gleicher Catid, wird der Makroaufruf abgewiesen.

NAME=poolname
Ordnet dem ISAM-Pool einen Namen zu. Zusammen mit Katalogkennung und Geltungsbereich ist der Pool so eindeutig identifizierbar.

„poolname“ kann 1-8 Zeichen lang sein; gültiger Zeichenvorrat: alle Buchstaben und Ziffern, sowie Sonderzeichen $, # und @; das erste Zeichen von „poolname“ muss ein Buchstabe oder eines der Sonderzeichen # oder @ sein.

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870)

= adr
adr ist die symbolische Adresse (der Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

PREFIX
wird nur in Verbindung mit MF=C oder MF=D ausgewertet und legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein Zeichen langes Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen sollen.

RESDNT
gibt an, ob die Seiten eines ISAM-Pools im Arbeitsspeicher resident gehalten werden sollen (im Sinn der Funktion $CSTAT):

= NO
gibt an, dass die Seiten des anzulegenden ISAM-Pools nicht resident gehalten werden sollen.

= YES
gibt an, dass die Seiten eines ISAM-Pools resident gehalten werden sollen. Zur Ausführung dieser Funktion ist ein Privileg für „Performanten Dateizugriff“ erforderlich.

Bei bereits existierenden ISAM-Pools wird der Aufruf abgewiesen, wenn ein Widerspruch zwischen dem RESDNT-Attribut des Pools und dem angeforderten RESDNT-Attribut besteht.

SCOPE
definiert den Geltungsbereich des ISAM-Pools.

= TASK
der ISAM-Pool kann nur vom aufrufenden Auftrag genutzt werden: er ist tasklokal.

= USERID
= USERGROUP
Die bis BS2000 OSD V6.0A vorhandenen Geltungsbereiche SCOPE=USERID und SCOPE=USERGROUP werden aus Kompatibilitätsgründen noch akzeptiert, intern jedoch auf SCOPE=HOST (taskübergreifender ISAM-Pool) abgebildet.

= HOST
Der ISAM-Pool kann von allen Aufträgen genutzt werden: er ist taskübergreifend.

Bei SCOPE=HOST wird der MODE-Operand ausgewertet; gleichzeitig wirkt sich SCOPE=HOST auf den WROUT-Operanden aus: die Voreinstellung WROUT=YES gilt für alle Dateien im ISAM-Pool und kann vom Benutzer nicht verändert werden.

SIZE
definiert die Größe des neu anzulegenden ISAM-Pools.

= STD
Der ISAM-Pool soll mit der bei Systeminstallation festgelegten Standardgröße angelegt werden.

Wenn der Parameter CREATION-MODE=ANY spezifiziert wurde, gilt Folgendes: Wird der Pool neu angelegt, so wird die Angabe zu SIZE ausgewertet wie oben beschrieben. Existiert der ISAM-Pool bereits, so wird die Größe des bestehenden ISAM-Pools übernommen. Die Angaben zu RESDNT und zu WROUT müssen dagegen mit den Attributen des existierenden ISAM-Pools übereinstimmen.

= zahl
bestimmt die Größe des anzulegenden ISAM-Pools in PAM-Seiten:
32 ≤ zahl ≤ 32767 bei Anlagen mit 31-Bit-Adressierung
32 ≤ zahl ≤ 2048 bei allen anderen Anlagen.
Ggf. stellt die vom Systemverwalter festgelegte maximale Größe des Benutzeradressraumes eine obere Grenze dar.
Ein ISAM-Pool, in dem Dateien gepuffert werden, die sowohl mit BLKCTRL=DATA2K als auch =DATA4K erstellt wurden, erhält dynamisch einen zweiten Extent, besteht also im Endeffekt aus einem 2K-Extent und einem 4K-Extent der Größe zahl.

Es ist zu beachten, dass mit der minimalen Größe von 32 PAM-Seiten nur Dateien mit einer Blockgröße bis höchstens (STD,6) verarbeitet werden können.

Mit der minimalen Poolgröße von 32 PAM-Seiten können jedoch nur Dateien mit einem Blockungsfaktor ≤ 6 (BLKSIZE ≤ (STD,6)) verarbeitet werden, es sei denn, die Dateien sind nur für lesende Zugriffe (d.h. MODUS=INPUT) eröffnet.

WROUT
legt für den Pool fest, ob geänderte Blöcke einer Datei sofort auf Platte geschrieben werden sollen:

Voreinstellung:
- WROUT=NO für einen tasklokalen ISAM-Pool (SCOPE=TASK)
- WROUT=YES für einen taskübergreifenden ISAM-Pool (SCOPE=USERID/USERGROUP/HOST)

= YES
Geänderte Blöcke werden sofort auf Platte geschrieben, unabhängig vom Wert des WROUT-Operanden im FILE- oder FCB-Aufruf für die zugehörige Datei.

= NO
gibt an, dass geänderte Blöcke nicht sofort auf Platte geschrieben werden müssen. Trotz einer Angabe von WROUT=NO bei CREPOOL wird jedoch ein geänderter Block sofort auf Platte geschrieben, wenn
- für die zugehörige Datei WROUT=YES (durch FILE oder FCB bzw. ADD-FILE-LINK) spezifiziert ist oder wenn
- für den Pool SCOPE≠TASK spezifiziert ist.

= UNCOND-NO
gibt an, dass geänderte Blöcke nicht sofort auf Platte geschrieben werden müssen, wobei die Einschränkungen von WROUT=NO nicht gelten:
– Auch für einen taskübergreifenden Pool (SCOPE=USERID/USERGROUP/HOST) werden geänderte Blöcke nicht sofort auf Platte geschrieben.
– Ein OPEN wird für Dateien, die mit SHARUPD=YES geöffnet werden sollen, nur dann durchgeführt, wenn im einem zugehörigen ADD-FILE-LINK-Kommando oder FILE-Makro explizit WRITE-IMMEDIATE=NO bzw. WROUT=NO spezifiziert wurde.

Ist WROUT für einen Pool nicht spezifiziert, so wird für einen taskslokalen Pool (SCOPE=TASK) WROUT=NO und für einen taskübergreifenden Pool (SCOPE=USERID,USERGROUP,HOST) WROUT=YES angenommen.

Returncodes

Die mit der C- oder D-Form des Makros generierten Feldnamen und EQU-Anweisungen für die Returncodes beginnen mit der Zeichenfolge DISC. Diese Zeichenfolge kann im ersten Zeichen durch PREFIX, in den Zeichen 2-4 durch MACID verändert werden.

Die Returncodes werden im Standardheader der Operandenliste hinterlegt.

Haupt-Returncode		**Bedeutung**
DISCOK	X'0000'	Makroaufruf war erfolgreich
DISCNPAR	X'0001'	auf die Operandenliste kann nicht zugegriffen werden
DISCNREM	X'0002'	der mit Catid bezeichnete Pubset befindet sich an einem Host-Rechner, auf dem eine BS2000-Version läuft, die ISAM-Pools nicht unterstützt
DISCNCAT	X'0003'	die Katalogkennung „catid“ ist im System nicht bekannt
DISCNACC	X'0004'	zum Pubset „catid“ besteht keine Verbindung
DISCINVN	X'0005'	der Poolname ist ungültig
DISCSPAC	X'0007'	der freie Adressraum reicht nicht aus, um einen Pool anzulegen (SIZE-Angabe zu groß)
DISCPLEX	X'0008'	Der angegebene ISAM-Pool existiert bereits: MODE=NEW wurde bereits von anderer Task angelegt MODE=ANY wurde bereits von derselben Task kreiert
DISCSYSE	X'000B'	während der Makrobearbeitung trat ein Systemfehler auf
DISCSIZE	X'000C'	SIZE-Angabe ist nicht zulässig
DISCINVW	X'000E'	WROUT-Angabe ist nicht zulässig
DISCINVS	X'000F'	SCOPE-Angabe ist nicht zulässig
DISCINVM	X'0010'	MODE-Angabe ist nicht zulässig
DISCPRIV	X'0011'	fehlendes Privileg für die Angabe RESDNT=YES
DISCPRES	X'0012'	die RESDNT-Angabe in der Parameterliste und des bereits existierenden Pools widersprechen sich.

Haupt-Returncode		**Bedeutung**
DISCPERR	X'0013'	Parameter-Fehler
DISCSPEX	X'0014'	Kontingent für ISAM-Pools überschritten
DISCRLNK	X'FFFF'	Makroaufruf konnte nicht ausgeführt werden (Linkage Fehler): Subreturncode1 auswerten!

DECFILE – Verschlüsselte Datei in unverschlüsselte Datei umwandeln

Makrotyp: S-Typ (E-Form/M-Form/L-Form/C-Form/D-Form) (siehe Seite 870)

Der DECFILE-Makroaufruf wandelt eine verschlüsselte Datei in eine unverschlüsselte Datei um (siehe Makro ENCFILE auf Seite 317).

Nach DECFILE sind alle Verschlüsselungsmerkmale (Verfahren und Kontrollstring für Crypto-Kennwort) im Katalogeintrag gelöscht.

Dateigenerationen

DECFILE kann nicht für einzelne Dateigenerationen angewendet werden, sondern nur für komplette Dateigenerationsgruppen. Innerhalb einer Dateigenerationsgruppe haben alle Generationen mit der Ausnahme von Bandgenerationen die gleichen Verschlüsselungsmerkmale wie der Gruppeneintrag.

Format

Operation	Operanden
DECFILE	,PATHNAM=<c-string 1..54: filename 1..54> / <var: char:54> ,EQUATES=YES / NO MF=L
	MF=D,PREFIX=D / <pre>
	MF=E,PARAM=<name 1..27>
	MF=C / M ,PREFIX=D / <pre> ,MACID=MAE / <macid>

Operandenbeschreibung

PATHNAM
Gibt die Datei an, die entschlüsselt werden soll. Das Crypto-Kennwort der Datei muss in der Crypto-Kennworttabelle der aufrufenden Task stehen.

=<c-string 1..54: filename 1..54>
Pfadname der Datei.

=<var: char:54>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 54 Byte, in dem der Pfadname der Datei abgelegt ist.

EQUATES
Gibt an, ob bei der Expansion des Parameterbereichs auch Equates für die Werte der Felder des Parameterbereichs generiert werden sollen.

= YES
Bei der Expansion des Parameterbereichs werden auch Equates für die Werte der Felder des Parameterbereichs generiert.

= NO
Bei der Expansion des Parameterbereichs werden keine Equates für die Werte der Felder des Parameterbereichs generiert.

Beispiel

```
:
MVC DECFMFC(YMAD#),DECFMFL
DECFILE MF=M,PREFIX=Y,PATHNAM='UMSATZ.3.QUARTAL.2004'
DECFILE MF=E,PARAM=DECFMFC
:
DECFMFC DECFILE MF=C,PREFIX=Y
DECFMFL DECFILE MF=L

:
```

Hinweise zur Programmierung

1. Vor der Generierung des Layouts des Parameterbereichs muss der Standard-Header aufgelöst werden.
2. Alle RESERVED-Felder des Parameterbereichs müssen mit binären Nullen gelöscht sein.
3. Bei allen Änderungen im Parameterbereich, die nicht mit Hilfe von GCs vorgenommen werden, ist der Aufrufer selbst für die Konsistenz des Parameterbereichs verantwortlich.
4. Beim nichtprivilegierten Aufruf (Funktionszustand TU) zeigt Register 1 auf den Parameterbereich.
5. Für die im Parameterbereich stehenden Namen erfolgt während der Funktionsausführung keine Umwandlung von Klein- in Großbuchstaben. Dagegen kann bei der GC-Expansion je nach Compiler-Einstellung eine Umwandlung von Klein- in Großbuchstaben erfolgen.

Hinweise zur Funktionsausführung

- Dateisperren und Dateischutzattribute, die den Schreibzugriff auf den Katalogeintrag oder auf den Inhalt einer Datei verbieten, verhindern damit auch das Umwandeln der Datei mit DECFILE.
- Das Umwandeln einer Datei mit DECFILE erfordert das Eigentumsrecht der aufrufenden Task an der Datei. Die Umwandlung erfolgt also, wenn:
  - Die Datei unter der Benutzerkennung der aufrufenden Task liegt.
  - Die aufrufende Task unter einer Benutzerkennung mit dem Privileg TSOS läuft.
  - Die Benutzerkennung der aufrufenden Task Mit-Eigentümer der Datei und die Datei nicht temporär ist.
- Zusätzliche Funktionen für Tasks mit Privileg TSOS:
  Wenn die aufrufende Task das Privileg TSOS hat, sind zusätzlich folgende Funktionen möglich:
  - Es können auch temporäre Dateien, die nicht zur aufrufenden Task, sondern zu einer anderen Task gehören, angegeben werden.
  - Es können auch temporäre Dateien auf einem anderen Pubset als dem Default-Pubset der Benutzerkennung angegeben werden. (Diese werden bei Beendigung der aufrufenden Task nicht automatisch gelöscht.)
- Die Umwandlung der verschlüsselten Datei wird mit SAT protokolliert.
  Das hierbei ausgegebene AUDIT-Attribut wird dem Katalogeintrag der umzuwandelnden Datei entnommen (siehe Kommando CREATE-FILE, Operand AUDIT, im Handbuch „Kommandos“ [3]).

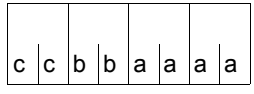
– RFA:
  DECFILE wird abgewiesen, wenn auf die umzuwandelnde Datei nur über RFA zugegriffen werden kann.

– Hilfsdatei:
  Beim Umwandeln mit DECFILE wird eine Hilfsdatei angelegt und bei Beendigung der Funktionsausführung automatisch gelöscht. In die Hilfsdatei wird der umgewandelte Dateiinhalt geschrieben. Die Hilfsdatei benötigt ebenso viel Plattenspeicherplatz wie die umzuwandelnde Datei.
  Der Dateiname der Hilfsdatei hat folgenden Aufbau:
  `S.DMS.<tsn>.<date><time>.CRYPTO`

Returncodes

Der Returncode wird im Standardheader der Parameterliste zurückgeliefert. Der Standardheader darf nicht im Read-only-Bereich liegen, sonst erfolgt Programmterminierung.

Standardheader:



Über die Ausführung des Makros DECFILE wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	**X'bb'**	**X'aaaa'**	**Erläuterung**
X'00'	X'00'	X'0000'	Kein Fehler
	X'01'	X'0554'	Format des Dateinamens unzulässig
	X'01'	X'0576'	a) Fehlerhafte Operandenkombination b) Nicht gelöschte UNUSED-Felder
	X'20'	X'0578'	Interner Fehler bei Überprüfung der Zugriffsrechte
	X'82'	X'0594'	Nicht genügend virtueller Speicher verfügbar
	X'20'	X'05C7'	Interner Fehler im DVS
	X'01'	X'05CB'	Fehlerhafter/unerlaubter erster Dateiname
	X'40'	X'05CF'	Kennwort nicht in Kennworttabelle
	X'40'	X'05FD'	Datei ist schreibgeschützt
	X'40'	X'0609'	Aktion für Systemdatei nicht erlaubt
	X'40'	X'0666'	Dateischutz verhindert Zugriff
X'01'	X'00'	X'066B'	Datei ist bereits entschlüsselt
X'00'	X'00'	X'066E'	Hilfsdatei verwenden
	X'01'	X'FFFF'	Falsche Funktionsnummer im Standardheader
	X'03'	X'FFFF'	Falsche Versionsnummer im Standardheader

DELAIX – Sekundärschlüssel einer ISAM-Datei löschen

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form) (siehe Seite 870)

Der Makro DELAIX löscht in einer NK-ISAM-Datei die vom Anwender ausgewählten oder alle Sekundärschlüssel.

Löschen eines Sekundärschlüssels bedeutet **nicht**, dass die Werte dieses Schlüssels in den Datensätzen gelöscht werden. Es werden vielmehr die zum Sekundärschlüssel gehörenden Sekundärindexblöcke gelöscht, sodass über diesen Sekundärschlüssel nicht mehr auf Datensätze zugegriffen werden kann.

Format

Operation	Operanden
DELAIX	,KEYNAME={(keyname1[,keyname2,...]) *ALL} ,{FILE=pfadname LINK=linkname} MF=L
	MF=E,PARAM={adr (r)}
	MF=D[,PREFIX=pre]
	MF=C[,PREFIX=pre][,MACID=macid]

Operandenbeschreibung

FILE = pfadname
Bezeichnet die NK-ISAM-Datei, deren im Operanden KEYNAME angegebenen Sekundärschlüssel gelöscht werden sollen, mit: <c-string 1..54: filename 1..54>

Die Angabe im FILE-Operanden wird ignoriert, wenn zugleich der LINK-Operand angegeben ist.

Pfadname bedeutet [:catid:][$userid.]dateiname

catid
Katalogkennung: falls nicht angegeben, wird die Default-Catid der Benutzerkennung angenommen.

userid
Benutzerkennung: falls nicht angegeben, wird die Benutzerkennung des SET-LOGON-PARAMETERS- bzw. des LOGON-Kommandos angenommen.

dateiname
vollqualifizierter Dateiname

KEYNAME
legt fest, welche Sekundärschlüssels gelöscht werden sollen.

= (keyname1[,keyname2,...])
Es werden alle Sekundärschlüssel gelöscht, deren Namen in der Liste aufgeführt sind. Die Sekundärschlüssel mit den Namen „keyname1“, „keyname2“ usw. müssen für die im Operanden FILE bzw. LINK angegebene Datei definiert sein. Über die Namen und Attribute aller für eine Datei vereinbarten Sekundärschlüssel kann sich der Anwender mit dem Makro SHOWAIX bzw. dem Kommando SHOW-INDEX-ATTRIBUTES informieren.
Die runden Klammern in dieser Angabe können weggelassen werden, wenn die Liste nur einen Namen enthält.

= *ALL
Es werden alle Sekundärschlüssel gelöscht, die für die im Operanden FILE bzw. LINK angegebene Datei definiert sind.

LINK = linkname
Legt den Dateikettungsnamen der Datei fest, deren im Operanden KEYNAME angegebenen Sekundärschlüssel gelöscht werden sollen.
„linkname“ darf bis zu acht Zeichen lang sein. Soll der Dateikettungsname über die Kommandoschnittstelle ansprechbar sein, muss er dem Datentyp <structured_name 1..8> entsprechen (siehe Handbuch „Kommandos“ [3]).

MACID
legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: MACID = IST

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870)

= adr
adr ist die symbolische Adresse (der Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

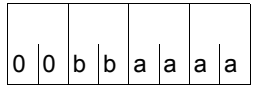
PREFIX
legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein Zeichen langes Präfix, mit dem die generierten Feldnamen und Equates beginnen sollen.

Returncodes

Standard-header:



Über die Ausführung des Makros DELAIX wird im Standardheader folgender Returncode übergeben (bb = SUBCODE1, aaaa = MAINCODE):

X'bb'	X'aaaa'	Erläuterung
X'00'	X'0000'	Funktion erfolgreich ausgeführt.
X'01'	X'0001'	Die Funktion konnte nicht ausgeführt werden: Die Operandenliste ist nicht verfügbar.
X'40'	X'0002'	Die Funktion konnte nicht ausgeführt werden: Sekundärschlüssel werden im fernen System nicht unterstützt (bei Makroaufruf über RFA).
X'40'	X'0003'	Die Funktion konnte nicht ausgeführt werden: Die angegebene Katalogkennung existiert nicht.
X'40'	X'0004'	Die Funktion konnte nicht ausgeführt werden: Auf den Katalog kann nicht zugegriffen werden.
X'01'	X'0005'	Die Funktion konnte nicht ausgeführt werden: Die Operandenliste enthält einen ungültigen Namen.
X'40'	X'0008'	Die Funktion konnte nicht ausgeführt werden: Der angegebene Sekundärschlüssel existiert nicht.
X'20'	X'000B'	Die Funktion konnte nicht ausgeführt werden: Systemfehler.
X'40'	X'000C'	Die Funktion konnte nicht ausgeführt werden: Der Benutzeradressraum ist zu klein.
X'40'	X'000E'	Die Funktion konnte nicht ausgeführt werden: Der Kontrollblock der Datei ist fehlerhaft.
X'40'	X'0012'	Die Funktion konnte nicht ausgeführt werden: Der ISAM-Pool ist überlastet.
X'40'	X'0016'	Die Funktion konnte nicht ausgeführt werden: Für KEYNAME wurde eine ungültige Anzahl von Schlüsselnamen angegeben.
X'01'	X'0017'	Die Funktion konnte nicht ausgeführt werden: In der Operandenliste wurde keine Datei spezifiziert.
X'40'	X'0018'	Die Funktion konnte nicht ausgeführt werden: Zum Zeitpunkt des Makroaufrufs ist SHARUPD=YES eingestellt.
X'40'	X'0019'	Die Funktion konnte nicht ausgeführt werden: Der Dateikettungsname ist fehlerhaft.
X'40'	X'0040'	Die Funktion konnte nicht ausgeführt werden: OPEN-Fehler.
X'40'	X'0041'	Die Funktion konnte nicht ausgeführt werden: CLOSE-Fehler.
X'40'	X'0044'	Die Funktion konnte nicht ausgeführt werden: Die Datei ist keine NK-ISAM-Datei.

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle „Standardheader“ auf Seite 873 entnommen werden.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

DELPOOL – ISAM-Pool löschen/freigeben

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form) (siehe Seite 870)

Mit dem Makroaufruf DELPOOL kann der Benutzer von ihm angelegte ISAM-Pools löschen bzw. die Verbindung zwischen seinem Auftrag und ISAM-Pools aufheben. Wird die Verbindung zwischen einem ISAM-Pool und dem letzten an ihn angeschlossenen Auftrag aufgehoben, wird der ISAM-Pool automatisch gelöscht.

Existiert der genannte ISAM-Pool nicht, wird der Makroaufruf mit einer Fehlermeldung abgewiesen.

Bevor ein ISAM-Pool bzw. die Verbindung zwischen Auftrag und ISAM-Pool gelöscht werden kann, müssen mit REMPLNK für die betroffenen Pools alle Einträge in den Pooltabellen gelöscht worden sein. Ist ein Pool noch über seinen Kettungsnamen mit einer Pooltabelle verknüpft, wird DELPOOL mit einer Fehlermeldung abgewiesen.

Hinweis

Taskübergreifende ISAM-Pools (SCOPE=HOST) werden bei der Dateieröffnung automatisch dateispezifisch in einem Data Space angelegt.
Die bis BS2000/OSD-BC V6.0A vorhandenen Geltungsbereiche SCOPE=USERID und SCOPE=USERGROUP werden aus Kompatibilitätsgründen noch akzeptiert, intern jedoch auf SCOPE=HOST (taskübergreifender ISAM-Pool) abgebildet. Für weitere Informationen zu ISAM-Pools in Data Spaces siehe Handbuch „Einführung in das DVS“ [1].

Format

Operation	Operanden
DELPOOL	MF=L,MODE={ SINGLE ,NAME=poolname[,CATID=catid][,SCOPE={ TASK | USERID | USERGROUP | HOST }] | ALL }
	MF=E,PARAM={ adr | (r) }
	MF=D[,PREFIX=pre]
	MF=C[,PREFIX=pre][,MACID=macid]

Operandenbeschreibung

MACID
wird nur in Verbindung mit MF=C ausgewertet und legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: MACID = ISD

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

MODE
gibt an, ob ein bestimmter ISAM-Pool gelöscht/freigegeben werden soll oder alle mit dem Auftrag verbundenen ISAM-Pools.

MODE = SINGLE
Es muss mindestens ein Poolname angegeben werden. Nur der durch NAME, CATID und SCOPE identifizierte ISAM-Pool wird gelöscht/freigegeben.

NAME = poolname
gibt den Namen an, mit dem der ISAM-Pool im CREPOOL-Makro angelegt wurde.

CATID = catid
gibt die Katalogkennung des Pubsets an, dem der ISAM-Pool im CREPOOL-Makroaufruf zugeordnet wurde.

Voreinstellung: die Default-Catid des Auftrags

SCOPE
gibt den Geltungsbereich des ISAM-Pools an, wie er im CREPOOL-Makroaufruf definiert wurde.

Alle Operandenwerte außer TASK werden nur noch aus Kompatibilitätsgründen unterstützt (siehe Hinweis auf Seite 258).

= TASK
der tasklokale ISAM-Pool „poolname“ wird gelöscht bzw. freigegeben, wenn kein Poolkettungsname mehr besteht; andernfalls wird der Makroaufruf mit Fehlermeldung abgebrochen.

= USERID
= USERGROUP
Die bis BS2000/OSD-BC V6.0A vorhandenen Geltungsbereiche SCOPE=USERID und SCOPE=USERGROUP werden aus Kompatibilitätsgründen noch akzeptiert, intern jedoch auf SCOPE=HOST (taskübergreifender ISAM-Pool) abgebildet.

= HOST
der taskübergreifende ISAM-Pool „poolname“ wird gelöscht bzw. freigegeben, wenn kein Poolkettungsname für den Auftrag mehr besteht (andernfalls: Abbruch mit Fehlermeldung).

MODE = ALL
gibt an, dass alle mit dem Auftrag verbundenen ISAM-Pools (tasklokal und taskübergreifend) gelöscht bzw. freigegeben werden.

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

= adr
adr ist die symbolische Adresse (der Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

PREFIX
wird nur in Verbindung mit MF=C oder MF=D ausgewertet und legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein Zeichen langes Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen sollen.

Returncodes

Die mit der C- oder D-Form des Makros generierten Feldnamen und EQU-Anweisungen für die Returncodes beginnen standardmäßig mit der Zeichenfolge DISD, die durch PREFIX und MACID geändert werden kann.

Die Returncodes werden im Standardheader der Operandenliste hinterlegt.

Maincode	**Bedeutung**
DISDOK X'0000'	Makroaufruf war erfolgreich
DISDNPAR X'0001'	auf die Operandenliste kann nicht zugegriffen werden
DISDNREM X'0002'	der mit Catid bezeichnete Pubset befindet sich an einem Host-Rechner, auf dem eine BS2000-Version läuft, die ISAM-Pools nicht unterstützt
DISDNCAT X'0003'	die Katalogkennung „catid“ ist im System nicht bekannt
DISDNACC X'0004'	zum Pubset „catid“ besteht keine Verbindung
DISDINVN X'0005'	der Poolname ist ungültig
DISDNANF X'0006'	es exsitiert kein ISAM-Pool mit diesem Namen, dieser Catid und diesem Geltungsbereich
DISDPUSE X'0009'	in der Pooltabelle des Auftrags bestehen noch Einträge mit Poolkettungs-namen für diesen ISAM-Pool
DISDSYSE X'000B'	während der Makrobearbeitung trat ein Systemfehler auf
DISDSPEX X'0014'	Kontingent für ISAM-Pools überschritten
DISDRNLK X'FFFF'	Makroaufruf konnte nicht ausgeführt werden (Linkage Fehler): Subreturncode1 auswerten!

DIV – Dateizugriff über virtuellen Adressraum

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Allgemeines

Unabhängig von der DIV-Funktion (Operand FCT) können bei einem DIV-Makro **alle** Operanden spezifiziert werden. Welche Operanden ausgewertet werden, hängt von der ausgewählten DIV-Funktion ab.
Zunächst werden alle Operanden in einer Übersicht dargestellt. Das Format und die Operanden, die bei den einzelnen Funktionen ausgewertet werden, werden pro Funktionseinheit beschrieben. Im Anschluss an die Formatübersicht werden kurz die Funktionen des DIV-Makros aufgelistet.

Operandenwerte, die nicht Adress- und nicht Register-Angaben sind, werden in der Operandenbeschreibung als „direkte Angabe“ bezeichnet.

Der Operandenwert „adr“ definiert eine symbolische Adresse, die in einer A-Konstanten abgelegt werden kann, d.h., die symbolische Adresse muss zum Übersetzungszeitpunkt berechnet werden können und darf nicht in einer DSECT enthalten sein.

Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

Parameterliste

Die Parameterliste des Makros enthält einen Header, dessen Felder beim Aufbau der Parameterliste mithilfe der L-Form automatisch versorgt werden.

Wird eine Parameterliste mit der D-Form oder C-Form dynamisch aufgebaut, ist sie zuvor durch eine mithilfe der L-Form erzeugten Parameterliste zu initialisieren. Nur auf diese Weise ist eine korrekte Versorgung des Header einer Parameterliste gewährleistet.

Wird in der folgenden Operandenbeschreibung auf Felder der Parameterliste Bezug genommen, so werden die Namen der Parameterliste angegeben, wie sie durch MF=D (ohne Angabe bei PREFIX) generiert werden.

Spezielle Parameter in der Parameterliste

Folgende Parameter sind Returnparameter, auf die nur direkt über die Parameterliste zugegriffen werden kann.

DIVPID

In DIVPID wird die Identifikation des OPEN zur Verfügung gestellt. Sie muss beim Aufruf weiterer DIV-Funktionen, die zu diesem OPEN gehören, in den Parameterlisten enthalten sein. Wird dieselbe Parameterliste benutzt, ist dies der Fall.

DIVPSIZE

In DIVPSIZE wird die logische Dateigröße in 4096-Byte-Seiten von OPEN zur Verfügung gestellt. DIVPSIZE enthält die Nummer der logisch letzten 4-KB-Seite (1 bedeutet, dass die erste Seite der Datei die logisch letzte Seite ist, 0 bedeutet, dass die Datei leer ist). DIVPSIZE kann zur Speicherbeschaffung für ein Fenster ausgewertet werden.

Wenn die Datei zuvor von der Zugriffsmethode UPAM bearbeitet wurde, ist es möglich, dass das logische Dateiende nicht an einer 4-KB-Seitengrenze liegt. In diesem Fall enthält DIVPSIZE den aufgerundeten Wert. Die letzte Halbseite vor dem logischen Dateiende erscheint dann in einem Fenster mit X'00' initialisiert.

Da die Datei durch SAVE logisch verkürzt oder verlängert werden kann, wird DIVPSIZE nach einem erfolgreichen SAVE-Aufruf aktualisiert. DIVPSIZE enthält dann die Nummer der logisch letzten 4-KB-Seite der Datei (1 bedeutet, dass die erste Seite der Datei die logisch letzte Seite ist, 0 bedeutet, dass die Datei leer ist).

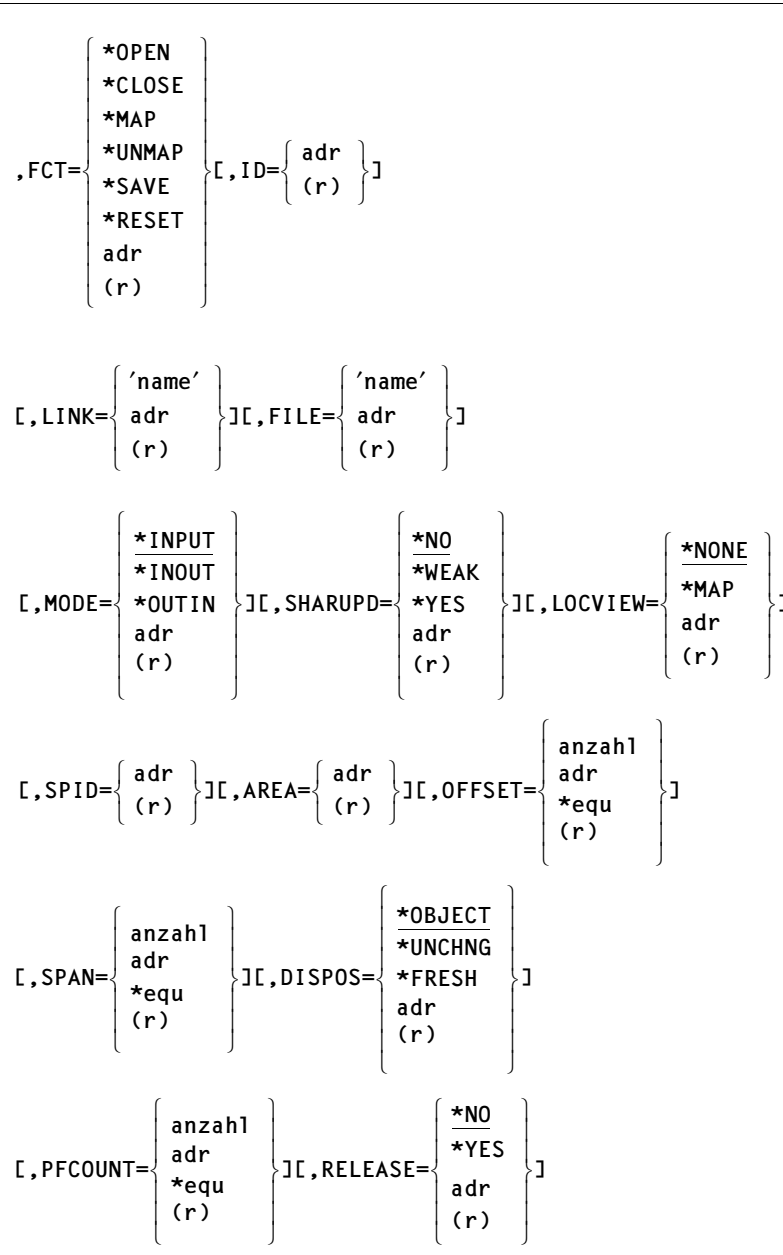
Modifikation der Dateieigenschaften durch Kommando oder Makro

Durch das Kommando ADD-FILE-LINK (oder durch den Makro FILE) kann der SHARUPD-Modus (NO | WEAK | YES) geändert werden.

Der OPEN-Modus (INPUT|INOUT|OUTIN) kann vom Kommando ADD-FILE-LINK bzw. vom Makro FILE **nicht** geändert werden.

Im Kommando ADD-FILE-LINK darf in den Operanden ACCESS-METHOD bzw. BLOCK-CONTROL-INFO keine Angabe erfolgen, die im Widerspruch zu den Dateistrukturattributen FILE_STRUC=PAM beziehungsweise BLK-CONTR=PAM steht. Das Gleiche gilt für die Operanden FCBTYPE beziehungsweise BLKCTRL des Makros FILE.

Formatübersicht

Operation	Operanden
DIV	

(Teil 1 von 2)

Operation	Operanden
	[,ENV={*HOST / *XCS / adr / (r)}]
	[,LARGE_FILE={*FORBIDDEN / *ALLOWED / adr / (r)}]
	MF=L
	MF=E,PARAM={adr / (r)}
	MF=D[,PREFIX={D / pre}]
	MF={C / M}[,PREFIX={D / pre}][,MACID={IVP / macid}]

(Teil 2 von 2)

Funktionsübersicht

Funktion	Kurzbeschreibung	siehe
FCT = *OPEN	DIV-/PAM-Datei eröffnen	Seite 266
FCT = *MAP	Fenster im Adressraum anlegen	Seite 274
FCT = *SAVE	Geänderte Fensterseiten in die Plattendatei zurückschreiben	Seite 281
FCT = *RESET	Änderungen in Fensterseiten rückgängig machen	Seite 286
FCT = *UNMAP	Fenster abbauen	Seite 292
FCT = *CLOSE	Plattendatei schließen	Seite 296

Funktion OPEN

Es wird eine Datei eröffnet und in die Parameterliste eine Kennung eingetragen (ID), die bei folgenden Aufrufen zu benutzen ist, um die OPEN-Zugehörigkeit der Aufrufe zu kennzeichnen.

Wenn für jeden zu einem DIV-OPEN gehörenden DIV-Aufruf dieselbe Parameterliste benutzt wird, ist die Kennung bereits eingetragen und braucht daher nicht beachtet zu werden.
Nach dem Aufruf enthält die Parameterliste die Dateigröße.

Von der Funktion OPEN werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet. Mithilfe weiterer Operanden kann jedoch die Parameterliste bereits für andere DIV-Funktionen vorbereitet werden.

Format FCT=*OPEN

Operation	Operanden
DIV	[,FCT={*OPEN | adr | (r)}] [,LINK={'name' | adr | (r)}] [,FILE={'name' | adr | (r)}] [,MODE={*INPUT | *INOUT | *OUTIN | adr | (r)}]

(Teil 1 von 2)

Operation	Operanden
	[,SHARUPD={ *NO / *WEAK / *YES / adr / (r) }]
	[,LOCVIEW={ *NONE / *MAP / adr / (r) }]
	[,ENV={ *HOST / *XCS / adr / (r) }]
	[,LARGE_FILE={ *FORBIDDEN / *ALLOWED / adr / (r) }]
	MF=L
	MF=E,PARAM={ adr / (r) }
	MF=D[,PREFIX={ D / pre }]
	MF={ C / M }[,PREFIX={ D / pre }][,MACID={ IVP / macid }]

(Teil 2 von 2)

Operandenbeschreibung

ENV
Beeinflusst die Verträglichkeit paralleler Eröffner in Abhängigkeit von ihrem Ablaufort (siehe dazu „Verträglichkeits-Matrix bei DIV-OPEN“ auf Seite 42).

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *HOST
Die maximal mögliche Parallelität ist auf Eröffner beschränkt, die im gleichen Host ablaufen.

= *XCS
Die Eröffner können in verschiedenen Hosts eines XCS-Rechnerverbunds ablaufen, ohne dass dadurch die Verträglichkeit eingeschränkt wird (z.B. können Schreiboperationen mit SHARUPD=*YES parallel ablaufen).

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für ENV enthält.

= (r)
Ist ein Register, das den Wert für ENVenthält.

FCT
Bestimmt die auszuführende DIV-Funktion.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *OPEN
Es wird eine Datei eröffnet und in die Parameterliste eine Kennung (ID) eingetragen. Diese Kennung muss bei den nächsten Aufrufen benutzt werden, um die OPEN-Zugehörigkeit der Aufrufe zu kennzeichnen.

Wird für jeden zu einem DIV-OPEN gehörenden DIV-Aufruf dieselbe Parameterliste benutzt, ist die Kennung bereits eingetragen und braucht daher nicht beachtet zu werden.

Nach dem OPEN-Aufruf enthält die Parameterliste die Kennung für den OPEN im Feld DIVPID.

Nach dem OPEN-Aufruf enthält die Parameterliste die Dateigröße im Feld `DIVPSIZE`

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die Funktion OPEN enthält (Wert `DIVPOPEN`, siehe Layout der der Parameterliste, Seite 302).

= (r)
Ist ein Register, das den Wert für die Funktion OPEN enthält.

FILE
Gibt den Dateinamen an.
Falls für den Operanden LINK ein Wert angegeben ist, wird eine FILE-Angabe nicht ausgewertet.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= '*name*'
Ist der in Hochkommata eingeschlossene Dateiname.
Länge des Dateinamens: 1-54 (mit catid) Zeichen.

= adr
Ist die symbolische Adresse eines 54 Byte langen Feldes, das den Dateinamen enthält.

= (r)
Ist ein Register, das die Adresse eines 54 Byte langen Feldes mit dem Dateinamen enthält.

LARGE_FILE
Der Operand LARGE_FILE bestimmt, ob die Dateigröße bei der Datenverarbeitung 32 GB überschreiten darf oder nicht (siehe Seite 109). Der Operand wird in die TFT (Task File Table) eingetragen und erst beim Öffnen der Datei mit OPEN ausgewertet.

Voreinstellung: LARGE_FILE = *FORBIDDEN

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= <u>FORBIDDEN</u>
Der Standardwert bedeutet, dass die Angaben im TU-FCB verwendet werden.

= ALLOWED
Die Datei wird als „große Datei“ angelegt: die Dateigröße kann 32 GB überschreiten.

= adr
Ist die Adresse eines 8 Byte langen Feldes, das den Wert für LARGE_FILE enthält.

= (r)
Ist ein Register, das die Adresse eines 8 Byte langen Feldes mit dem Wert für LARGE_FILE enthält.

LINK
Spezifiziert den Dateikettungsnamen (LINK-Name) der Datei.

Über Dateikettungsname/TFT werden Programm und Datei miteinander verknüpft (zu Dateikettungsname/TFT siehe Handbuch „Einführung in das DVS“ [1]).

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= 'name'
Ist der in Hochkommata eingeschlossene Dateikettungsname.
„name“ darf bis zu acht Zeichen lang sein. Soll der Dateikettungsname über die Kommandoschnittstelle ansprechbar sein, muss er dem Datentyp <structured_name 1..8> entsprechen (siehe Handbuch „Kommandos“ [3]).

= adr
Ist die Adresse eines 8 Byte langen Feldes, das den Dateikettungsnamen enthält.

= (r)
Ist ein Register, das die Adresse eines 8 Byte langen Feldes mit dem Dateikettungsnamen enthält.

LOCVIEW
Mit dem Operand LOCVIEW wird spezifiziert, ob Seiten zum MAP-Zeitpunkt in ein Fenster eingelesen werden oder erst nach einem Zugriff auf eine Seite.

Voreinstellung: LOCVIEW = *NONE

Der Operand LOCVIEW ist nur für Fenster wirksam, die mit DISPOS=*OBJECT angelegt sind.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *MAP
Bei der Definition eines Fensters (Funktion FCT=*MAP) werden alle Dateiseiten bereits zum MAP-Zeitpunkt in das Fenster eingelesen. Bei einem SHARUPD=*WEAK-Leser verhindert DIV, dass ein paralleler SHARUPD=*WEAK-Schreiber gleichzeitig mit SAVE die Datei verändern kann.

= *NONE
Eine Seite wird beim ersten Zugriff auf sie aus der Datei in das Fenster gelesen (Voreinstellung).

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das einen Wert für LOCVIEW enthält (`DIVPLNON` | `DIVPLMAP`; siehe Layout der Parameterliste, Seite 302).

= (r)
Ist ein Register, das einen Wert für LOCVIEW enthält.

MACID
Legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

= IVP
Voreinstellung: MACID=IVP

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

MODE
Bestimmt den OPEN-Modus (siehe Abschnitt „Multi-User-Betrieb“ auf Seite 41):
Der OPEN-Modus kann **nicht** durch ein ADD-FILE-LINK-Kommando verändert werden (Returncode DIVPICFS (INCOMPATIBLE_FILE_SPEC)).

Voreinstellung: MODE=*INPUT

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *INPUT
Die Datei kann nur gelesen werden. Die Funktion SAVE ist daher bei diesem OPEN-Modus nicht möglich.

Unabhängig vom SHARUPD-Modus sind parallele INPUT-Eröffnungen möglich, auch von der Zugriffsmethode UPAM.

Die Datei muss existieren, d.h. sie muss schon einmal mit OUTIN eröffnet worden sein.

= *INOUT
Die Datei kann modifiziert werden, d.h. in die Datei kann mit der DIV-Funktion SAVE geschrieben werden.

Die Datei muss existieren, d.h. sie muss schon einmal mit OUTIN eröffnet worden sein.

Parallele Eröffnungen sind möglich in Abhängigkeit vom SHARUPD-Modus.

= *OUTIN
Die Datei wird neu erstellt, d.h. nach OPEN ist die Datei „leer“. Danach kann in die Datei geschrieben werden. Wie bei MODE=*INOUT kann mit der SAVE-Funktion in die Datei geschrieben werden.

Parallele Eröffnungen sind möglich in Abhängigkeit vom SHARUPD-Modus.

Bei einem Multi-User-Betrieb (SHARUPD=*WEAK|*YES) muss ein MODE=*OUTIN-Eröffner immer der erste Eröffner sein. Sonst wird der OPEN zurückgewiesen.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das einen Wert für MODE enthält (`DIVPINPT` | `DIVPINOT` | `DIVPOUTI`; siehe Layout der Parameterliste, Seite 302).

= (r)
Ist ein Register, das einen Wert für MODE enthält.

PARAM
Bezeichnet die Adresse der Operandenliste. Der Operand wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

PREFIX
Legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

= D
Voreinstellung: PREFIX=D

= pre
„pre“ ist ein ein Zeichen langes Präfix, mit dem die generierten Feldnamen und Equates beginnen sollen.

SHARUPD
Steuert den parallelen Zugriff mehrerer Anwender (siehe auch Handbuch „Einführung in das DVS“ [1], Abschnitt „Multi-User-Betrieb an einem Rechner“).

Voreinstellung: SHARUPD = *NO

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *NO
Mehrere parallele Leser (MODE=*INPUT) **oder** ein Schreiber (MODE=*INOUT oder *OUTIN) sind erlaubt.

= *WEAK
Mehrere parallele Leser (MODE=*INPUT) **und** ein Schreiber (MODE=*INOUT oder *OUTIN) sind erlaubt. Der Inhalt der im Fenster eines Lesers (MODE=*INPUT) enthaltenen Daten ist also abhängig von Änderungen durch einen parallelen Schreiber (MODE=*INOUT | *OUTIN) und vom Einlesezeitpunkt einer Seite in das Fenster. Leser und Schreiber müssen sich daher koordinieren.
Die Angabe von LOCVIEW=*MAP bewirkt, dass die Konsistenz der Daten in einem Fenster auch bei fehlender Koordinierung der Tasks nicht durch einen parallelen Schreiber zerstört wird.

= *YES
Mehrere Schreiber dürfen eine Datei eröffnen.

Hinweise

Die Konsistenz der Daten in der Datei wird in diesem Fall von DIV nicht sichergestellt, sondern muss vielmehr vom Anwender selbst gewährleistet werden, z.B. durch Serialisierung von SAVE-Aufrufen durch den Anwender.

Bei jedem Aufruf des Allocators wird die Dateigröße überprüft.
Wenn bei dieser Überprüfung eine Dateigröße ≥ 32 GB ermittelt wird und im zugehörigen FCB das Attribut LARGE_FILE=*FORBIDDEN bzw. in der TFT das Attribut EXCEED-32GB=*FORBIDDEN gesetzt ist, wird die Verarbeitung abgebrochen. DIV liefert in diesem Fall den Returncode `X'00400030'` in seiner eigenen Parameterliste DIV(I).

Durch ein ADD-FILE-LINK-Kommando kann der SHARUPD-Modus verändert werden.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das einen Wert für SHARUPD enthält (Wert `DIVPSNO` | `DIVPSWEA` | `DIVPSYES`; siehe Layout der Parameterliste, Seite 302).

= (r)
Ist ein Register, das einen Wert für SHARUPD enthält.

Hinweis

Die Parameter DIVPID und DIVPSIZE sind Returnparameter, auf die nur direkt über die Parameterliste zugegriffen werden kann. Weitere Informationen hierzu siehe Abschnitt „Parameterliste“, Seite 262.

Funktion MAP

Die Funktion MAP erzeugt ein Fenster in einem Adressraum (Programmraum oder Datenraum). Ein Fenster ist einem Dateibereich oder einer ganzen Datei zugeordnet.

Der Adressraum muss vor dem Aufruf der MAP-Funktion allokiert sein (explizit durch REQM, implizit durch Binder-Lader). Eine Freigabe des Adressraums, in dem ein Fenster enthalten ist, ist erst nach dem Abbau eines Fensters (durch UNMAP) möglich.

Im Fensterbereich darf keine READ-ONLY-Seite und – zum MAP-Zeitpunkt – keine I/O-fixed Seite enthalten sein, also keine Seite, auf der eine Ein-/Ausgabe läuft (z.B. asynchrone Ein-/Ausgabe durch UPAM während MAP).

Der Adressraum darf nicht sharable sein.

Bei Ausführung der MAP-Funktion wird von DIV sichergestellt, dass alle Seiten der Datei, die vom Fenster repräsentiert werden, allokiert sind. Liegt ein Fenster oder ein Teil eines Fensters hinter dem physikalischen Dateiende, werden die fehlenden Seiten allokiert. Für einen INPUT-Eröffner wird keine Allokierung durchgeführt.

Mit dem Operanden DISPOS kann spezifiziert werden, ob im Fenster die Seiten der Datei erscheinen sollen oder ob die Daten des Adressraums erhalten bleiben sollen.

Von der Funktion MAP werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet.

Format FCT=*MAP

Operation	Operanden
DIV	[,FCT={*MAP / adr / (r)}][,ID={adr / (r)}][,SPID={adr / (r)}] [,AREA={adr / (r)}][,OFFSET={anzahl / adr / *equ / (r)}][,SPAN={anzahl / adr / *equ / (r)}] [,DISPOS={*OBJECT / *UNCHNG / adr / (r)}][,PFCOUNT={anzahl / adr / *equ / (r)}]
	MF=L
	MF=E,PARAM={adr / (r)}
	MF=D[,PREFIX={D / pre}]
	MF={C / M}[,PREFIX={D / pre}][,MACID={IVP / macid}]

Operandenbeschreibung

AREA
AREA gibt die Anfangsadresse des Fensters innerhalb des Adressraums an, der durch den Operanden SPID spezifiziert ist (Datenraum bzw. Programmraum).

Vor dem Aufruf der MAP-Funktion muss der Adressraum allokiert sein (Makro REQM, DSPSRV, Binder-Lader). Er kann nicht vor Abbau des Fensters (UNMAP) freigegeben werden.

Das Fenster muss an einer 4-KB-Seitengrenze liegen. Seine Länge ist durch den Operanden SPAN bestimmt.

Eine Seite im virtuellen Adressraum darf nur **einem** Fenster angehören. Versuche, eine bereits für ein Fenster vergebene Seite für ein weiteres Fenster zu benutzen, werden zurückgewiesen.

Bei der MF=L-Form darf die Anfangsadresse des Fensters nur durch eine symbolische Adresse angegeben werden.

= adr
Ist die 4 Byte lange Anfangsadresse des Fensters (symbolische Adresse).

= (r)
Ist ein Register, das die Anfangsadresse des Fensters enthält.

DISPOS
Mit DISPOS wird bestimmt, welche Daten im Fenster nach MAP sichtbar sein sollen: die im Adressraum unveränderten Daten (wie vor MAP) oder die Daten der entsprechenden Seiten der Datei.

Voreinstellungen: DISPOS = *OBJECT

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *OBJECT
Im Fenster erscheinen die Seiten der Datei. Seiten hinter der logisch letzten Dateiseite erscheinen mit X'00' gefüllt.

= *UNCHNG
Die Seiten im Fenster werden nicht durch Seiten der Datei ersetzt sondern behalten ihren Inhalt.

Ein mit DISPOS=*UNCHNG definiertes Fenster kann dazu benutzt werden, die entsprechenden Seiten der Datei bei Ausführung der Funktion SAVE mit dem Inhalt der Seiten des virtuellen Adressraums zu initialisieren (siehe SAVE-Funktion und Beschreibung der logischen Dateiverlängerung im Handbuch „Einführung in das DVS“ [1]).

DISPOS=*FRESH darf bei FCT=*MAP nicht spezifiziert werden.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das einen Wert für DISPOS enthält (`DIVPOBJ | DIVPUNCH`; siehe Layout der Parameterliste, Seite 302).

= (r)
Ist ein Register, das einen Wert für DISPOS enthält.

FCT
Bestimmt die auszuführende DIV-Funktion.
Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *MAP
Mit der DIV-Funktion MAP wird ein Fenster in einem Adressraum (Programmraum oder Datenraum) angelegt. Nähere Beschreibung siehe Seite 274.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die Funktion MAP enthält (Wert `DIVMAP`, siehe Layout der Parameterliste, Seite 302).

= (r)
Ist ein Register, das den Wert für die Funktion MAP enthält.

ID
Mit ID wird der OPEN spezifiziert, für den die entsprechende DIV-Funktion ausgeführt werden soll.

Wird dieselbe Parameterliste benutzt wie bei OPEN, ist die Angabe von ID nicht erforderlich, da sich dann die Identifikation des OPEN bereits in der Parameterliste befindet. Die Identifikation ist im Feld `DIVPID` der Parameterliste enthalten.
Wird eine andere Parameterliste benutzt wie bei OPEN, kann durch ID die Identifikation zur Verfügung gestellt und in die neue Parameterliste übertragen werden (mit der Form MF=M des Makros DIV).

ID kann nicht mit der MF=L-Form angegeben werden.

= adr
Ist die symbolischen Adresse eines 8 Byte langen Feldes, das die Identifikation enthält.

= (r)
Ist ein Register mit der Adresse des 8 Byte langen Feldes.

MACID
Zu MACID siehe Beschreibung beim Format FCT=*OPEN, Seite 271.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

OFFSET
OFFSET legt zusammen mit SPAN den Dateibereich fest, für den das Fenster angelegt wird.

– OFFSET legt den Beginn des Dateibereiches fest. Mit OFFSET wird angegeben, ab welchem Block (ab welcher 4-KB-Seite) der Dateibereich beginnt.
– SPAN gibt an, aus wie vielen 4-KB-Blöcken der Dateibereich (Bereichslänge) bestehen soll.

Der durch OFFSET und SPAN definierte Dateibereich wird dem Fenster im virtuellen Adressraum zugeordnet.

Voreinstellung: OFFSET = 0

Bei OFFSET = 0 entspricht die erste Fensterseite der ersten Seite der Datei. Die Datei wird ab Dateibeginn in der Länge von SPAN in das Fenster eingelesen.

Falls für SPAN kein Wert angegeben ist (oder SPAN=0), wird die Fenstergröße so gewählt, dass die letzte Seite des Fensters der logisch letzten Seite der Datei entspricht. Falls weder OFFSET noch SPAN spezifiziert ist, wird also die Fenstergröße so gewählt, dass die gesamte Datei bis zur logisch letzten Seite im Fenster enthalten ist.

Ist die Datei leer und ist für SPAN der Standardwert gegeben, wird der MAP-Aufruf zurückgewiesen.

SPAN und OFFSET können so gewählt werden, dass Seiten, die hinter dem **logischen** Ende der Datei liegen, im Fenster erscheinen. Seiten hinter dem logischen Dateiende erscheinen im Fenster mit X'00' gefüllt.

SPAN und OFFSET können auch so gewählt werden, dass Seiten, die hinter dem **physikalischen** Dateiende liegen, im Fenster erscheinen. In diesem Fall werden durch MAP bei OPEN OUTIN | INOUT so lange weitere Blöcke für die Datei allokiert, bis die der letzten Fensterseite entsprechende Dateiseite allokiert ist.

Hinweis

Durch MAP kann eine Datei physikalisch verlängert werden, durch SAVE kann eine Datei logisch verlängert (und verkürzt) werden. Das logische Dateiende wird durch MAP nicht verändert (nur durch SAVE).

Für ein und denselben OPEN darf eine Dateiseite nur **einem** Fenster zugeordnet sein. Eine Dateiseite kann jedoch mehreren Fenstern zugeordnet sein, wenn diese zu verschiedenen OPEN gehören.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= anzahl
Gibt den ersten Block des im virtuellen Adressraum abzubildenden Dateibereichs an. Der Wert für OFFSET ist begrenzt durch die maximale Größe einer Datei in 4-KB-Seiten minus 1:

$0 \leq$ anzahl $\leq$ 8388606 bei LARGE_FILE=*FORBIDDEN

$0 \leq$ anzahl $\leq$ 1073741823 bei LARGE_FILE=*ALLOWED

= adr
Ist die symbolische Adresse eines 4 Byte langen Feldes, das den numerischen Wert (binär) für die Angabe des ersten Blocks des im virtuellen Adressraum abzubildenden Dateibereichs enthält.

= *equ
Ist ein Equate, das den numerischen Wert für die Angabe des ersten Blocks des im virtuellen Adressraum abzubildenden Dateibereichs darstellt. Dem Namen des Equate muss das Zeichen „*“ vorausgehen.

= (r)
Ist ein Register, das den numerischen Wert für die Angabe des ersten Blocks des im virtuellen Adressraum abzubildenden Dateibereichs enthält.

PARAM
Zu PARAM siehe Beschreibung beim Format FCT=*OPEN, Seite 272.

PREFIX
Zu PREFIX siehe Beschreibung beim Format FCT=*OPEN, Seite 272.

PFCOUNT
Wenn aufseiten eines Fensters sequenziell (in aufsteigender Folge) zugegriffen wird, können Page-Fault-Interrupts durch Angabe von PFCOUNT verringert werden. Wird als Folge eines Page-Fault-Interrupt eine Seite aus der Datei in ein Fenster eingelesen und ist PFCOUNT für das Fenster spezifiziert, so werden die folgenden Seiten mit einer einzigen Leseoperation zusätzlich eingelesen, bis die durch PFCOUNT spezifizierte Anzahl von Seiten, das Fensterende oder eine bereits eingelesene Seite erreicht ist.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= anzahl
Gibt an, wie viele Seiten zusätzlich eingelesen werden sollen.
$0 \leq$ anzahl ≤ 15

= adr
Ist die Adresse eines 4 Byte langen Feldes, das den numerischen Wert (binär) für die Anzahl der Seiten enthält.

= *equ
Ist ein Equate, das den numerischen Wert darstellt. Dem Namen des Equate muss das Zeichen „*“ vorausgehen.

= (r)
Die Angabe eines Registers mit dem numerischen Wert.

SPAN
SPAN definiert zusammen mit OFFSET den Dateibereich, für den das Fenster angelegt wird. Der mit SPAN und OFFSET definierte Dateibereich wird dem Fenster im virtuellen Adressraum zugeordnet.

Voreinstellung: SPAN = 0

Zur Beschreibung von SPAN siehe Beschreibung beim Operanden OFFSET.
Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= anzahl
Gibt die Länge des Dateibereiches in 4-KB-Blöcken an. Der Wert für SPAN ist begrenzt durch die maximale Adressraumgröße (2 GB) in 4-KB-Seiten.
$0 \leq$ anzahl $\leq$ 524287

= adr
Ist die symbolische Adresse eines 4 Byte langen Feldes, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) angibt.

= *equ
Ist ein Equate, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) angibt. Dem Namen des Equates muss das Zeichen „*“ vorausgehen.

= (r)
Ist ein Register, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) angibt.

SPID
Spezifiziert den Adressraum (Daten- oder Programmraum), in dem das Fenster angelegt werden soll.

Voreinstellung: SPID = 0

Wird SPID nicht angegeben oder ist SPID = 0, wird das Fenster im Programmraum angelegt. Sonst wird mit SPID ein Datenraum angegeben.
SPID kann nicht mit der MF=L-Form angegeben werden.

= adr
Ist die symbolische Adresse eines 8 Byte langen Feldes, das die SPID enthält.

= (r)
Register, das die Adresse eines 8 Byte langen Feldes enthält,das die SPID enthält.

Funktion SAVE

Die Funktion SAVE schreibt modifizierte Fensterseiten in die Datei, wenn sie in einem durch SPAN und OFFSET definierten Dateibereich liegen.

Wenn im definierten Bereich modifizierte Fensterseiten enthalten sind, die hinter dem logischen Dateiende liegen, wird die Datei verlängert: die letzte modifizierte Seite bestimmt das neue logische Dateiende. Bei einer Dateiverlängerung werden auch nicht modifizierte Fensterseiten in die Datei geschrieben, wenn sie sich zwischen dem bisherigen und dem neuen logischen Dateiende befinden. Ist für Teile des Bereichs, um den eine Datei verlängert wird, kein Fenster definiert, werden für diese Teilbereiche keine Seiten in die Datei geschrieben. Die entsprechenden Dateiseiten haben dann undefinierten Inhalt (zur logischen Dateiverlängerung siehe auch Handbuch „Einführung in das DVS“ [1]).

Wenn sich die logisch letzte Seite der Datei in einem mit DISPOS=*UNCHNG definierten Fenster befindet und wenn keine Bedingung vorliegt, die zu einer Dateiverlängerung führt, wird die Datei verkürzt, wenn noch nicht auf die logisch letzte Seite zugegriffen wurde, d.h. wenn sich die logisch letzte Seite noch im Initialzustand befindet. Die Verkürzung geht so weit, bis eine Seite gefunden ist, die nicht mehr im Initialzustand ist, oder bis eine Seite gefunden ist, die schon einmal durch SAVE in die Datei geschrieben wurde, oder bis eine Seite erreicht ist, die nicht zu einem mit DISPOS=*UNCHNG definierten Fenster gehört (zur logischen Dateiverkürzung siehe auch Handbuch „Einführung in das DVS“ [1]).

Eine durch SAVE in die Datei geschriebene Seite gilt nicht mehr als modifiziert, d.h. dass sie bei einem erneuten SAVE nur dann in die Datei geschrieben wird, wenn sie nach dem vorausgegangenen SAVE erneut modifiziert wurde.

Die SAVE-Funktion ist nicht erlaubt, wenn die Datei mit MODE=*INPUT eröffnet wurde.

Von der Funktion SAVE werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet.

Format FCT=*SAVE

Operation	Operanden
DIV	[,FCT={*SAVE / adr / (r)}][,ID={adr / (r)}] [,OFFSET={anzahl / adr / *equ / (r)}][,SPAN={anzahl / adr / *equ / (r)}] MF=L
	MF=E,PARAM={adr / (r)}
	MF=D[,PREFIX={D / pre}]
	MF={C / M}[,PREFIX={D / pre}][,MACID={IVP / macid}]

Operandenbeschreibung

FCT
Bestimmt die auszuführende DIV-Funktion.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *SAVE
Mit der DIV-Funktion SAVE werden geänderter Fensterseiten in die Datei auf Platte (siehe auch Seite 281) zurückgeschrieben.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die Funktion SAVE enthält (`DIVPSAVE`; siehe Layout der Parameterliste, Seite 302).

= (r)
Ist ein Register, das den Wert für die Funktion SAVE enthält.

ID
Mit ID wird der OPEN angegeben, für den die Funktion SAVE ausgeführt werden soll.

Wird dieselbe Parameterliste benutzt wie bei OPEN, ist die Angabe von ID nicht erforderlich, da sich dann die Identifikation des OPEN bereits in der Parameterliste befindet. Die Identifikation ist im Feld DIVPID der Parameterliste enthalten.

Wird eine andere Parameterliste benutzt wie bei OPEN, kann durch ID die Identifikation zur Verfügung gestellt und in die neue Parameterliste übertragen werden (mit der Form MF=M des DIV-Makros).

ID kann nicht mit der MF=L-Form angegeben werden.

= adr
Ist die symbolischen Adresse eines 8 Byte langen Feldes, das die Identifikation enthält.

= (r)
Ist ein Register mit der Adresse eines 8 Byte langen Feldes, das die Identifikation enthält.

MACID
Zu MACID siehe Beschreibung beim Format FCT=*OPEN, Seite 271.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

OFFSET
OFFSET legt zusammen mit SPAN den Dateibereich fest, auf den sich SAVE bezieht.
– OFFSET spezifiziert den Beginn des Dateibereiches. Mit OFFSET wird angegeben, ab welchem Block (ab welcher 4-KB-Seite) der Dateibereich beginnt.
– SPAN gibt an, aus wie vielen 4-KB-Blöcken der Dateibereich (Bereichslänge) bestehen soll.

Auf alle Fensterseiten des mit OFFSET und SPAN definierten Dateibereichs wird die SAVE-Funktion angewendet.

Voreinstellung: OFFSET = 0

Falls für SPAN kein Wert angegeben ist (oder SPAN = 0), wird der Bereich so gewählt, dass die letzte Seite des letzten für den OPEN definierten Fensters im Bereich enthalten ist. Falls weder OFFSET noch SPAN angegeben ist, werden also alle Seiten aller für den OPEN definierten Fenster berücksichtigt.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= anzahl
Gibt den ersten Block des im virtuellen Adressraum abzubildenden Dateibereichs an. Der Wert für OFFSET ist begrenzt durch die maximale Größe einer Datei in 4-KB-Seiten minus 1:

$0 \leq$ anzahl $\leq$ 8388606 bei LARGE_FILE=*FORBIDDEN

$0 \leq$ anzahl $\leq$ 1073741823 bei LARGE_FILE=*ALLOWED

= adr
Ist die symbolische Adresse eines 4 Byte langen Feldes, das den numerischen Wert für die Angabe des ersten Blocks des im virtuellen Adressraum abzubildenden Dateibereichs enthält (binär).

= *equ
Ist ein Equate, das den numerischen Wert für die Angabe des ersten Blocks des im virtuellen Adressraum abzubildenden Dateibereichs darstellt. Dem Namen des Equate muss das Zeichen „*“ vorausgehen.

= (r)
Ist ein Register, das den numerischen Wert für die Angabe des ersten Blocks des im virtuellen Adressraum abzubildenden Dateibereichs enthält.

PARAM
Zu PARAM siehe Beschreibung beim Format FCT=*OPEN, Seite 272.

PREFIX
Zu PREFIX siehe Beschreibung beim Format FCT=*OPEN, Seite 272.

SPAN
SPAN definiert zusammen mit OFFSET den Dateibereich, auf den sich SAVE bezieht.

Voreinstellung: SPAN = 0

Zur Beschreibung von SPAN siehe Beschreibung beim Operanden OFFSET.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= anzahl
Gibt die Länge des Dateibereiches in 4-KB-Blöcken an. Der Wert für SPAN ist begrenzt durch die maximale Größe einer Datei in 4-KB-Seiten:

$0 \leq$ anzahl $\leq$ 8388607 bei LARGE_FILE=*FORBIDDEN

$0 \leq$ anzahl $\leq$ 1073741824 bei LARGE_FILE=*ALLOWED

= adr
Ist die symbolische Adresse eines 4 Byte langen Feldes, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) enthält.

= *equ
Ist ein Equate, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) angibt. Dem Namen des Equates muss das Zeichen „*“ vorausgehen.

= (r)
Ist ein Register, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) enthält.

Spezielle Parameter in der Parameterliste

Der folgende Parameter ist ein Returnparameter, auf den nur direkt über die Parameterliste zugegriffen werden kann. Es wird der Name der Parameterliste angegeben, der durch MF=D (ohne Angabe eines PREFIX) generiert wird:

DIVPSIZE

Da die Datei durch SAVE logisch verkürzt oder verlängert werden kann, enthält DIVPSIZE nach einem erfolgreichen SAVE-Aufruf die Nummer der logisch letzten 4-KB-Seite der Datei (1: die erste Seite der Datei ist die logisch letzte Seite; 0: die Datei ist leer).

Funktion RESET

Die Funktion RESET macht Modifikationen von Fensterseiten rückgängig, die einem durch SPAN und OFFSET definierten Dateibereich angehören.

Dies geschieht so, dass eine modifizierte Seite in ihren Initialzustand gebracht wird, was zur Folge hat, dass sie bei einem Zugriff aus der Datei gelesen wird. Gehört die Seite zu einem mit DISPOS=*UNCHNG definierten Fenster, wird sie bei einem Zugriff nur dann aus der Datei gelesen, wenn sie schon einmal durch SAVE in die Datei geschrieben wurde, sonst wird sie mit X'00' initialisiert.

Durch die Angabe RELEASE=*YES werden alle Seiten des definierten Bereichs (nicht nur modifizierte Seiten) in ihren Initialzustand gebracht. Dies bietet die Möglichkeit, einen durch einen parallelen Schreiber erzeugten neuen Zustand der Dateiseiten in Fenstern erscheinen zu lassen.

Von der Funktion RESET werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet.

Format FCT=*RESET

Operation	Operanden
DIV	[,FCT={*RESET / adr / (r)}] [,ID={adr / (r)}] [,OFFSET={anzahl / adr / *equ / (r)}] [,SPAN={anzahl / adr / *equ / (r)}] [,RELEASE={*NO / *YES / adr / (r)}] MF=L
	MF=E,PARAM={adr / (r)}
	MF=D[,PREFIX={D / pre}]
	MF={C / M}[,PREFIX={D / pre}][,MACID={IVP / macid}]

Operandenbeschreibung

FCT
Bestimmt die auszuführende DIV-Funktion.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *RESET
Mit der DIV-Funktion RESET werden Modifikationen von Fensterseiten rückgängig gemacht, die einem durch SPAN und OFFSET definierten Dateibereich angehören (ausführliche Beschreibung siehe Seite 286).

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die Funktion RESET enthält (Feld `DIVPRES`, siehe Layout der der Parameterliste, Seite 302).

= (r)
Ist ein Register, das den Wert für die Funktion RESET enthält.

ID
Mit ID wird der OPEN spezifiziert, für den die Funktion RESET ausgeführt werden soll.

Wird dieselbe Parameterliste benutzt wie bei OPEN, ist die Angabe von ID nicht erforderlich, da sich dann die Identifikation des OPEN bereits in der Parameterliste befindet.

Wird eine andere Parameterliste benutzt wie bei OPEN, kann durch ID die Identifikation zur Verfügung gestellt und in die neue Parameterliste übertragen werden (mit der Form MF=M des Makros DIV).

Bei der Form MF=L kann ID nicht angegeben werden.

= adr
Ist die symbolischen Adresse eines 8 Byte langen Feldes, das die Identifikation enthält.

= (r)
Ist ein Register mit der Adresse des 8 Byte langen Feldes.

MACID
Zu MACID siehe Beschreibung beim Format FCT=*OPEN, Seite 271.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

OFFSET
OFFSET spezifiziert zusammen mit SPAN den Dateibereich im Fenster, für den die Fensterseiten in ihren Initialzustand zurückgesetzt werden sollen.
– OFFSET spezifiziert den Beginn des Dateibereiches. Mit OFFSET wird angegeben, ab welchem Block (ab welcher 4-KB-Seite) der Dateibereich beginnt.
– SPAN gibt an, aus wie vielen 4-KB-Blöcken der Dateibereich (Bereichslänge) bestehen soll.

Auf alle Fensterseiten des mit OFFSET und SPAN definierten Dateibereichs wird die RESET-Funktion angewendet.

Voreinstellung: OFFSET = 0

Falls für SPAN kein Wert angegeben ist (oder SPAN = 0), wird der Bereich so gewählt, dass die letzte Seite des letzten Fensters im Bereich enthalten ist. Falls weder OFFSET noch SPAN spezifiziert ist, werden also alle Seiten aller für den OPEN definierten Fenster berücksichtigt.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= anzahl
Gibt die Länge des Dateibereiches in 4-KB-Blöcken an. Der Wert für OFFSET ist begrenzt durch die maximale Größe einer Datei in 4-KB-Seiten minus 1:

$0 \leq$ anzahl $\leq$ 8388606 bei LARGE_FILE=*FORBIDDEN

$0 \leq$ anzahl $\leq$ 1073741823 bei LARGE_FILE=*ALLOWED

= adr
Ist die symbolische Adresse eines 4 Byte langen Feldes, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) enthält.

= *equ
Ist ein Equate, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) angibt. Dem Namen des Equates muss das Zeichen „*“ vorausgehen.

= (r)
Ist ein Register, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) enthält.

PARAM
Zu PARAM siehe Beschreibung beim Format FCT=*OPEN, Seite 272.

PREFIX
Zu PREFIX siehe Beschreibung beim Format FCT=*OPEN, Seite 272.

RELEASE
Durch den Operanden RELEASE wird angegeben, ob nur modifizierte Seiten in den Initialzustand gebracht werden sollen.

Voreinstellung: RELEASE=*NO

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *NO
Alle modifizierten Seiten werden in den Initialzustand gebracht.

Dies hat zur Folge, dass bei einem Zugriff auf eine solche Seite die entsprechende Seite aus der Datei gelesen wird, wenn sie in einem mit DISPOS=*OBJECT definierten Fenster liegt. Liegt die Seite in einem mit DISPOS=*UNCHNG definierten Fenster, wird sie bei einem Zugriff mit X'00' initialisiert, wenn sie noch nicht durch SAVE in die Datei geschrieben wurde, sonst wird sie bei einem Zugriff aus der Datei gelesen.

Eine Seite im Initialzustand, die hinter der logisch letzten Seite der Datei liegt, erscheint bei einem Zugriff immer mit X'00' initialisiert.

= *YES
Alle Fensterseiten – geänderte wie nicht geänderte – im angegebenen Bereich werden in den Initialzustand gebracht, mit den oben beschriebenen Folgen.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das einen Wert für RELEASE enthält (`DIVPRNO` | `DIVPRYES`; siehe Layout der Parameterliste, Seite 302).

= (r)
Ist ein Register, das einen Wert für RELEASE enthält.

SPAN
SPAN definiert zusammen mit OFFSET den Dateibereich, auf den sich die RESET-Funktion bezieht.

Voreinstellung: SPAN = 0

Zur Beschreibung von SPAN siehe auch Beschreibung beim Operanden OFFSET.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= anzahl
Gibt die Länge des Dateibereiches in 4-KB-Blöcken an. Der Wert für SPAN ist begrenzt durch die maximale Größe einer Datei in 4-KB-Seiten:

$0 \leq$ anzahl $\leq$ 8388607 bei LARGE_FILE=*FORBIDDEN

$0 \leq$ anzahl $\leq$ 1073741824 bei LARGE_FILE=*ALLOWED

= adr
Ist die symbolische Adresse eines 4 Byte langen Feldes, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) enthält.

= *equ
Ist ein Equate, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) angibt. Dem Namen des Equates muss das Zeichen „*“ vorausgehen.

= (r)
Ist ein Register, das die Länge des Dateibereiches in 4-KB-Blöcken (binär) enthält.

Funktion UNMAP

Die Funktion UNMAP baut ein durch MAP erzeugtes Fenster wieder ab.

UNMAP verursacht keine Änderungen von Dateiseiten.

Der Operand DISPOS bestimmt den Zustand, in dem die Fensterseiten hinterlassen werden:

– Falls DISPOS=*UNCHNG spezifiziert ist, haben die Seiten im Fenster nach UNMAP denselben Inhalt wie vor UNMAP – aus der Sicht des Programms. Dies bedeutet, dass alle Seiten, die bei einem Zugriff vor UNMAP aus der Datei gelesen würden, zum UNMAP-Zeitpunkt aus der Datei gelesen werden müssen, da ein Einlesen aus der Datei nach dem Abbau des Fensters nicht mehr möglich ist.

– Falls DISPOS=*FRESH spezifiziert ist, werden alle Seiten des Fensters in ihren Initialzustand gebracht. Sie erscheinen bei einem Zugriff nach UNMAP mit X'00' initialisiert.

Von der Funktion UNMAP werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet.

Format FCT=*UNMAP

Operation	Operanden
DIV	[,FCT={*UNMAP / adr / (r)}] [,SPID={adr / (r)}] [,AREA={adr / (r)}] [,DISPOS={*FRESH / *UNCHNG / adr / (r)}] MF=L
	MF=E,PARAM={adr / (r)}
	MF=D[,PREFIX={D / pre}]
	MF={C / M}[,PREFIX={D / pre}][,MACID={IVP / macid}]

Operandenbeschreibung

AREA
AREA spezifiziert die Anfangsadresse des Fensters innerhalb des Adressraums, der durch SPID spezifiziert ist (Datenraum bzw. Programmraum, wenn kein Datenraum spezifiziert ist).

Bei der Form MF=L darf die Anfangsadresse des Fensters nur durch eine symbolische Adresse angegeben werden.

= adr
Ist die 4 Byte lange Anfangsadresse des Fensters (symbolische Adresse).

= (r)
Ist ein Register, das die Anfangsadresse des Fensters enthält.

DISPOS
Mit DISPOS wird bestimmt, mit welchem Inhalt die Fensterseiten hinterlassen werden sollen.

Voreinstellung: DISPOS = *FRESH

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *UNCHNG
Die Seiten erscheinen wie vor dem Abbau des Fensters.

Diese Variante kann zu einem ungünstigen Performanceverhalten führen, wenn Seiten aus der Datei gelesen werden, auf die nach UNMAP nicht mehr zugegriffen wird.

= *FRESH
Alle Seiten des Fensters werden in ihren Initialzustand gebracht. Sie erscheinen mit X'00' initialisiert.

Wird für FCT=*UNMAP eine Parameterliste benutzt, die bereits von einer vorausgegangenen Funktion *MAP verwendet wurde, ist zu beachten, dass bereits ein Wert für den Operanden DISPOS eingesetzt wurde. Der Operand DISPOS ist eventuell neu zu bestimmen, insbesondere ist zu beachten, dass DISPOS=*OBJECT für UNMAP keinen gültigen Operandenwert darstellt.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das einen Wert für DISPOS enthält (`DIVPFRSH` | `DIVPUNCH`; siehe Parameterliste, Seite 302).

= (r)
Ist ein Register, das einen Wert für DISPOS enthält.

FCT
Bestimmt die auszuführende DIV-Funktion.

= *UNMAP
Die Funktion UNMAP baut ein durch MAP erzeugtes Fenster wieder ab (ausführliche Funktionsbeschreibung siehe Seite 292.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die Funktion UNMAP enthält (Feld `DIVUNMP`, s. Layout der Parameterliste Seite 302).

= (r)
Ist ein Register, das den Wert für die Funktion UNMAP enthält.

MACID
Zu MACID siehe Beschreibung beim Format FCT=*OPEN, Seite 271.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

PARAM
Zu PARAM siehe Beschreibung beim Format FCT=*OPEN, Seite 272.

PREFIX
Zu PREFIX siehe Beschreibung beim Format FCT=*OPEN, Seite 272.

SPID
Gibt den Adressraum (Daten- oder Programmraum) an, in dem das Fenster liegt.

SPID kann nicht mit der MF=L-Form angegeben werden.

= adr
Ist die symbolische Adresse eines 8 Byte langen Feldes, das die Identifikation des Adressraums enthält.

= (r)
Ist ein Register, das die Adresse eines 8 Byte langen Feldes enthält, in dem die Identifikation des Adressraumes (Programm-/Datenraum) steht.

Funktion CLOSE

Die Funktion CLOSE schließt die Datei.

Falls für den OPEN noch Fenster definiert sind, werden sie mit Standardwerten für die Operanden abgebaut.

Von der Funktion CLOSE werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet.

Format FCT=*CLOSE

Operation	Operanden
DIV	[,FCT={ *CLOSE / adr / (r) }] [,ID={ adr / (r) }] MF=L
	MF=E,PARAM={ adr / (r) }
	MF=D[,PREFIX={ D / pre }]
	MF={ C / M }[,PREFIX={ D / pre }][,MACID={ IVP / macid }]

Operandenbeschreibung

FCT
Bestimmt die auszuführende DIV-Funktion.

= *CLOSE
Die Funktion CLOSE schließt die Datei.
Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die Funktion CLOSE enthält (Feld `DIVPCLS`, siehe Layout der Parameterliste, Seite 302).

= (r)
Ist ein Register, das den Wert für die Funktion CLOSE enthält.

ID
Mit ID wird der OPEN spezifiziert, für den die Funktion CLOSE ausgeführt werden soll.

Wird dieselbe Parameterliste benutzt wie bei OPEN, ist die Angabe von ID nicht erforderlich, da sich dann die Identifikation des OPEN bereits in der Parameterliste befindet. Die Identifikation ist in DIVPID (siehe Parameterliste) enthalten.
Wird eine andere Parameterliste benutzt wie bei OPEN, kann durch ID die Identifikation zur Verfügung gestellt und in die neue Parameterliste übertragen werden (mit der Form MF=M des Makros DIV).

ID kann nicht mit der MF=L-Form angegeben werden.

= adr
Ist die symbolischen Adresse eines 8 Byte langen Feldes, das die Identifikation enthält.

= (r)
Register mit der Adresse eines 8 Byte langen Feldes, das die Identifikation enthält.

MACID
Zu MACID siehe Beschreibung beim Format FCT=*OPEN, Seite 271.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

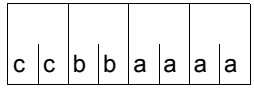
PARAM
Zu PARAM siehe Beschreibung beim Format FCT=*OPEN, Seite 272.

PREFIX
Zu PREFIX siehe Beschreibung beim Format FCT=*OPEN, Seite 272.

Returncodes

Returncodes werden im Header der Parameterliste abgelegt. Die DIV-spezifischen Main-Returncodes werden in nachfolgender Tabelle erläutert. Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, sowie der Aufbau des Standardheaders, sind auf Seite 873 beschrieben.

Standard-header:



Über die Ausführung des Makros DIV wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Funktion erfolgreich ausgeführt
	X'01'	X'0001'	Die Parameterliste ist zumindest nicht in ihrer vollen Länge zugreifbar. Falls auch auf den Header(teil) der Parameterliste nicht zugegriffen werden kann oder die Parameterliste nicht auf Wortgrenze ausgerichtet ist, wird das Programm mit einer Fehlermeldung abgebrochen.
	X'01'	X'0002'	Die durch AREA (bei MAP oder UNMAP) spezifizierte Fensteradresse liegt nicht an einer 4-KB-Seitengrenze.
	X'01'	X'0003'	Bei MAP, SAVE und RESET: Die Bereichsgröße (SPAN) führt (evtl. im Zusammenhang mit dem spezifizierten OFFSET) zu einer zu großen Plattenadresse. Bei MAP zusätzlich möglich: Die Fenstergröße (SPAN) ist größer als 2 GB.
	X'01'	X'0004'	Bei MAP, SAVE und RESET: Der Wert für OFFSET entspricht einer zu großen Plattenadresse.
	X'01'	X'0005'	Bei MAP: Der Wert für DISPOS ist weder *OBJECT noch *UNCHNG. Bei UNMAP: Der Wert für DISPOS is weder *FRESH noch *UNCHNG.
	X'01'	X'0006'	Der Wert für PFCOUNT liegt nicht im Bereich von 0 bis 15 (MAP).
	X'01'	X'0007'	Der Wert für RELEASE ist weder *NO noch *YES (RESET).
	X'01'	X'0008'	Der Wert für MODE (OPEN) ist weder *INPUT noch *INOUT noch *OUTIN.
	X'01'	X'0009'	Der Wert für SHARUPD (OPEN) ist weder *NO noch *WEAK noch *YES.
	X'01'	X'000A'	Der Wert für LOCVIEW (OPEN) ist weder *NONE noch *MAP.
	X'01'	X'000B'	Der Wert für ENV (OPEN) ist weder *HOST noch *XCS.
	X'01'	X'000C'	Der Wert für LARGE_FILE (OPEN) ist weder *ALLOWED noch *FORBIDDEN.
	X'01'	X'000D'	DIV wird auf SPARC-HSI nicht unterstützt.

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'0014'	DIV-Aufrufe aus TPR sind nicht erlaubt.
	X'40'	X'0015'	Von der (allgemeinen) OPEN-Behandlung des DMS wurde ein Fehler erkannt. Der DMS-Fehlercode ist im Feld DIVPDMSC enthalten.
	X'40'	X'0016'	Von der (allgemeinen) CLOSE-Behandlung des DMS wurde ein Fehler erkannt. Der DMS-Fehlercode ist im Feld DIVPDMSC enthalten.
	X'40'	X'0017'	Die Datei ist bereits eröffnet. Die OPEN-Anforderung wird zurückgewiesen, weil MODE-Wert und/oder SHARUPD-Wert der aktuellen OPEN-Anforderung und der bereits existierenden Eröffnung keine parallele Bearbeitung gestatten.
	X'40'	X'0018'	Der OPEN wird auf Grund folgender Situationen zurückgewiesen: – Im Kommando ADD-FILE-LINK wurde ein OPEN MODE spezifiziert, der von DIV nicht unterstützt wird (z.B. OUTPUT). – Im Kommando ADD-FILE-LINK wurde ein anderer OPEN MODE spezifiziert als vom Programm. – Im Kommando ADD-FILE-LINK wurde ein unzulässiger Wert für ACCESS-METHOD spezifiziert. – Für eine neue Datei (MODE=*OUTIN) wurde im Kommando ADD-FILE-LINK der Operand BUFFER-LENGTH explizit mit einem Wert ungleich 2 spezifiziert. – Im Kommando ADD-FILE-LINK wurde der Operand BLOCK-CONTROL-INFO fehlerhaft spezifiziert oder es wird versucht, eine existierende Datei zu eröffnen, die nicht die Eigenschaft BLKCTRL=NO besitzt.
	X'40'	X'0019'	Es ist weder ein Linkname noch ein Dateiname angegeben (OPEN).
	X'40'	X'001A'	Die zu eröffnende Datei befindet sich auf einer Shared Private Disk (SPD). Dateien auf Shared Private Disk (SPD) werden von DIV nicht bearbeitet.
	X'40'	X'001B'	Ein Fenster mit dem Attribut DISPOS=*UNCHNG ist nicht für eine mit MODE=*INPUT eröffnete Datei erlaubt (MAP).
	X'40'	X'001C'	Die Privilegierung (USER oder SYSTEM) des Eröffners (OPEN) unterscheidet sich von der Privilegierung des Rufers der Funktion MAP, UNMAP, SAVE, RESET oder CLOSE.
	X'40'	X'001D'	Die bei einer der Funktionen MAP, UNMAP, SAVE, RESET oder CLOSE zur Kennzeichnung des OPEN mitgegebene ID ist DIV nicht (mehr) bekannt. Möglicherweise ist die Parameterliste von der Parameterliste des OPEN verschieden und ID wurde nicht versorgt oder die Datei wurde bereits geschlossen.
	X'40'	X'001E'	SPID wurde angegeben, kennzeichnet jedoch keinen Datenraum für das Fenster oder einen Datenraum, auf den der Rufer nicht zugreifen darf (MAP, UNMAP).

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'001F'	Zumindest ein Teil des für ein Fenster spezifizierten Adressraums ist bereits von einem existierenden Fenster belegt (MAP).
	X'40'	X'0020'	Zumindest ein Teil des für ein Fenster spezifizierten Dateibereichs ist bereits in einem Fenster des Eröffners abgebildet (MAP).
	X'40'	X'0021'	SPAN ist nicht spezifiziert (bzw. SPAN=0) und kann zur Definition eines Fensters (MAP) von DIV nicht bestimmt werden, weil entweder – die Datei leer ist oder – OFFSET hinter die logisch letzte Seite zeigt.
	X'40'	X'0022'	Der Fensterbereich enthält mehrere Speicherklassen (z.B. Klasse-5-Speicher und Klasse-6-Speicher) (MAP).
	X'40'	X'0023'	Eine Seite im virtuellen Adressraum soll zu einer Fensterseite gemacht werden und ist (für eine Ein/Ausgabe) fixiert (MAP).
	X'40'	X'0024'	Eine für ein Fenster vorgesehene Seite im virtuellen Adressraum ist resident (MAP).
	X'40'	X'0025'	Eine für ein Fenster vorgesehene Seite im virtuellen Adressraum ist eine READ-ONLY Seite (MAP).
	X'40'	X'0026'	Zumindest ein Teil des für ein Fenster definierten Adressraums ist nicht allokiert (es wurde z. B. kein REQM durchgeführt) (MAP).
	X'40'	X'0027'	Der für ein Fenster spezifizierte Adressraum ist sharable (MAP).
	X'40'	X'0028'	Der für ein Fenster spezifizierte Adressraum ist nicht für den nicht-privilegierten Anwender zugreifbar und der Rufer ist nicht privilegiert (MAP).
	X'40'	X'0029'	Eine DIV-interne Tabelle kann nicht angelegt werden wegen fehlendem Benutzeradressraum (MAP).
	X'40'	X'002A'	Die Datei kann nicht (physikalisch) verlängert werden (MAP), weil – entweder kein Plattenbereich mehr für den Benutzer zur Verfügung steht oder – für die Datei eine Sekundärallokierung nicht vorgesehen ist (Operand SPACE... SEC-ALLOC im Kommando CREATE-FILE oder MODIFY-FILE-ATTRIBUTES).
	X'40'	X'002B'	Beim Lesen eines Blocks trat ein Fehler auf (MAP, UNMAP).
	X'40'	X'002C'	Beim Schreiben eines Blocks trat ein Fehler auf (SAVE).
	X'40'	X'002D'	Das durch AREA (und SPID) definierte Fenster existiert nicht für den aktuellen OPEN (UNMAP).
	X'40'	X'002E'	Für den durch OFFSET und SPAN definierten Datenbereich existiert kein Fenster (SAVE, RESET).
	X'40'	X'002F'	SAVE ist für einen Leser nicht erlaubt.

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'0030'	Beim Zugriff auf eine Datei im Modus SHARUPD=YES wurde festgestellt, dass die Dateigröße den Wert von 32 GB übersteigt, beim OPEN für diese Datei wurde aber ein Überschreiten von 32 GB nicht erlaubt.
	X'80'	X'003C'	Das Subsystem DIV wurde durch ein Kommando gestoppt. Weitere OPEN werden daher abgewiesen.
	X'80'	X'003E'	Zum Aufbau interner Tabellen von DIV benötigter Systemadressraum steht nicht zur Verfügung (OPEN, MAP).
	X'20'	X'0046'	DIV-interner Fehler.
	X'20'	X'0047'	Vermutlich auf Grund eines DIV-internen Fehlers wurde vergeblich auf die Freigabe eines gesperrten Betriebsmittels gewartet.

Erläuterungen zu den Returncodes

Returncodes werden im Header der Parameterliste abgelegt:
– Der Main Returncode in einem Halbwort mit dem Namen DIVPMRET.
– Der Subcode1 in einem Byte mit dem Namen DIVPSR1.
  Subcode1 beschreibt Fehlerklassen, die es dem Aufrufer emöglichen, auf Fehlerklassen zu reagieren (vgl. Tabelle Seite 873). Der Aufrufer kann sich sowohl am Maincode als auch am Subcode1 orientieren.
– Der Subcode2 wird z.Zt. nicht verwendet. Er ist immer null (X'00').

In folgenden Fällen können die Returncodes nicht im Header abgelegt werden:
– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Das aufrufende Programm wird dann mit einer Fehlermeldung beendet (siehe Abschnitt „Fehlermeldungsschlüssel im DVS“ auf Seite 875) und das STXIT Ereignis „nicht behebbarer Programmfehler“ erzeugt.

Die mit der C- oder D-Form des Makros generierten Feldnamen und EQU-Anweisungen für die Returncodes beginnen standardmäßig mit der Zeichenfolge DIVP, die durch PREFIX und MACID geändert werden kann.

Layout der Parameterliste

Von einem DIV-Makroaufruf mit MF=D wird folgende Parameterliste abgesetzt:

```
           DIV   MF=D
1          MFCHK MF=D,PREFIX=D,MACID=IVP,PARAM=,
1                SVC=126,DMACID=IVP,DNAME=IVPLIST,SUPPORT=(C,D,E,L,M)
2 DIVPLIST DSECT ,
2                *,##### PREFIX=D, MACID=IVP #####
1          #INTF REFTYPE=REQUEST,INTNAME=DIV,INTCOMP=002
1 *
1 DIVPPA   DS    0F          BEGIN of PARAMETERAREA
1          FHDR  MF=(C,DIVP),EQUATES=YES
2          DS    0A
2 DIVPFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
2 *
2 DIVPIFID DS    0A              0   INTERFACE IDENTIFIER
2 DIVPFCTU DS    AL2             0   FUNCTION UNIT NUMBER
2 *                                  BIT 15    HEADER FLAG BIT,
2 *                                  MUST BE RESET UNTIL FURTHER NOTICE
2 *                                  BIT 14-12 UNUSED, MUST BE RESET
2 *                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
2 DIVPFCT  DS    AL1             2   FUNCTION NUMBER
2 DIVPFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
2 *
2 DIVPRET  DS    0A              4   GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 DIVPSRET DS    0AL2            4   SUB RETURN CODE
2 DIVPSR2  DS    AL1             4   SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 DIVPR2OK EQU   X'00'               All correct, no additional info
2 DIVPR2NA EQU   X'01'               Successful, no action was necessary
2 DIVPR2WA EQU   X'02'               Warning, particular situation
2 DIVPSR1  DS    AL1             5   SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A    X'00'           FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B    X'01' - X'1F'   PARAMETER SYNTAX ERROR
2 * CLASS C    X'20'           INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D    X'40' - X'7F'   NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E    X'80' - X'82'   WAIT AND RETRY
2 *
2 DIVPRFSP EQU   X'00'               FUNCTION SUCCESSFULLY PROCESSED
```

```
2 DIVPRPER EQU   X'01'               PARAMETER SYNTAX ERROR
2 *  3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 DIVPRFNS EQU   X'01'               CALLED FUNCTION NOT SUPPORTED
2 DIVPRFNA EQU   X'02'               CALLED FUNCTION NOT AVAILABLE
2 DIVPRVNA EQU   X'03'               INTERFACE VERSION NOT SUPPORTED
2 *
2 DIVPRAER EQU   X'04'               ALIGNMENT ERROR
2 DIVPRIER EQU   X'20'               INTERNAL ERROR
2 DIVPRCAR EQU   X'40'               CORRECT AND RETRY
2 *  2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 DIVPRECR EQU   X'41'               SUBSYSTEM (SS) MUST BE CREATED
2 *                                  EXPLICITELY BY CREATE-SS
2 DIVPRECN EQU   X'42'               SS MUST BE EXPLICITELY CONNECTED
2 *
2 DIVPRWAR EQU   X'80'               WAIT FOR A SHORT TIME AND RETRY
2 DIVPRWLR EQU   X'81'                    "     LONG       "
2 DIVPRWUR EQU   X'82'               WAIT TIME IS UNCALCULABLY LONG
2 *                                  BUT RETRY IS POSSIBLE
2 *  2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 DIVPRTNA EQU   X'81'               SS TEMPORARILY NOT AVAILABLE
2 DIVPRDH  EQU   X'82'               SS IN DELETE / HOLD
2 *
2 DIVPMRET DS    0AL2           6    MAIN RETURN CODE
2 DIVPMR2  DS    AL1            6    MAIN RETURN CODE 2
2 DIVPMR1  DS    AL1            7    MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XXYYYY')
2 *
2 DIVPRLNK EQU   X'FFFF'             LINKAGE ERROR / REQ. NOT PROCESSED
2 DIVPFHL  EQU   8              8    GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *                          DIV-FUNCTIONS:
1 DIVPOPEN EQU   1           OPEN
1 DIVPCLS  EQU   2           CLOSE
1 DIVPMAP  EQU   3           MAP
1 DIVPUNMP EQU   4           UNMAP
1 DIVPSAVE EQU   5           SAVE
1 DIVPRES  EQU   6           RESET
1 *                          DIV-MAINCODES:
1 DIVPPNAC EQU   1           PARLIST NOT ACCESSIBLE
1 DIVPIWAD EQU   2           INVALID WINDOW ADDRESS
1 DIVPISPA EQU   3           INVALID SPAN
1 DIVPIOFF EQU   4           INVALID OFFSET
1 DIVPIDSP EQU   5           INVALID DISPOS
1 DIVPIPFC EQU   6           INVALID PFCOUNT
1 DIVPIREL EQU   7           INVALID RELEASE
1 DIVPIOM  EQU   8           INVALID OPEN MODE
1 DIVPISUM EQU   9           INVALID SHARUPD MODE
```

```
1 DIVPILVM EQU   10          INVALID LOCVIEW MODE
1 DIVPIENV EQU   11          INVALID LOCKENV
1 DIVPILRF EQU   12          INVALID LARGE_FILE
1 DIVPNSPA EQU   13          DIV NOT SUPPORTED ON SPARC-HSI
1 DIVPPRVC EQU   20          PRIVILEGED DIV CALL
1 DIVPDOER EQU   21          DMS OPEN ERROR
1 DIVPDCER EQU   22          DMS CLOSE ERROR
1 DIVPICOM EQU   23          INCOMPATIBLE OPEN MODE
1 DIVPICFS EQU   24          INCOMPATIBLE FILE SPEC
1 DIVPNLNF EQU   25          NEITHER LINK NOR FILE
1 DIVPPD   EQU   26          PRIVATE DISK
1 DIVPDOI  EQU   27          DISPOS OPEN INCONSISTENCY
1 DIVPPRVI EQU   28          PRIV INCONSISTENCY
1 DIVPWRID EQU   29          WRONG ID
1 DIVPSIDU EQU   30          SPID UNDEFINED
1 DIVPSPOV EQU   31          SPACE OVERLAP
1 DIVPFOV  EQU   32          FILE OVERLAP
1 DIVPUSNP EQU   33          UNDEF SPAN NOT POSSIBLE
1 DIVPINHM EQU   34          INHOMOG MEM
1 DIVPPFIX EQU   35          PAGE FIXED
1 DIVPRESP EQU   36          RESIDENT PAGE
1 DIVPROP  EQU   37          READ ONLY PAGE
1 DIVPMNA  EQU   38          MEM NOT ALLOC
1 DIVPSHRM EQU   39          SHARABLE MEMORY
1 DIVPPRSP EQU   40          PRIVILEGED SPACE
1 DIVPNOAS EQU   41          NO ADDRESS SPACE
1 DIVPALER EQU   42          ALLOC ERROR
1 DIVPRDER EQU   43          READ ERROR
1 DIVPWRER EQU   44          WRITE ERROR
1 DIVPWNF  EQU   45          WINDOW NOT FOUND
1 DIVPNWIR EQU   46          NO WINDOW IN RANGE
1 DIVPSAVN EQU   47          SAVE NOT ALLOWED
1 DIVPLFNS EQU   48          LARGE_FILE NOT SPECIFIED
1 DIVPSSS  EQU   60          SUBSYSTEM STOPPED
1 DIVPSOR  EQU   61          SHORTAGE OF RESOURCES
1 DIVPIERR EQU   70          INTERNAL ERROR
1 DIVPTOUT EQU   71          TIME RUNOUT
1 *
1 DIVPID   DS    XL8         ID
1 DIVPLARF DS    AL1         LARGE_FILE
1 DIVPFRBD EQU   0           LARGE_FILE=FORBIDDEN
1 DIVPALWD EQU   1           LARGE_FILE=ALLOWED
1 DIVPUNUS DS    AL1         UNUSED BYTE
1 DIVPDMSC DS    H           DMS-CODE
1 DIVPLINK DS    CL8         LINK
1 DIVPFILE DS    CL54        FILE
1 DIVPOMOD DS    AL1         MODE
1 DIVPINPT EQU   1           MODE=INPUT
```

```
1 DIVPINOT EQU   2           MODE=INOUT
1 DIVPOUTI EQU   3           MODE=OUTIN
1 DIVPSUPD DS    AL1         SHARUPD
1 DIVPSNO  EQU   1           SHARUPD=NO
1 DIVPSWEA EQU   2           SHARUPD=WEAK
1 DIVPSYES EQU   3           SHARUPD=YES
1 DIVPLOCV DS    AL1         LOCVIEW
1 DIVPLNON EQU   1           LOCVIEW=NONE
1 DIVPLMAP EQU   2           LOCVIEW=MAP
1 DIVPDISP DS    AL1         DISPOS
1 DIVPOBJ  EQU   1           DISPOS=OBJECT
1 DIVPUNCH EQU   2           DISPOS=UNCHNG
1 DIVPFRSH EQU   3           DISPOS=FRESH
1 DIVPREL  DS    AL1         RELEASE
1 DIVPRNO  EQU   1           RELEASE=NO
1 DIVPRYES EQU   2           RELEASE=YES
1 DIVPLENV DS    AL1         LOCKENV
1 DIVPHOST EQU   1           LOCKENV=HOST
1 DIVPXCS  EQU   2           LOCKENV=XCS
1 DIVPSIZE DS    F           SIZE
1 DIVPSPID DS    XL8         SPID
1 DIVPAREA DS    A           AREA
1 DIVPOFFS DS    F           OFFSET
1 DIVPSPAN DS    F           SPAN
1 DIVPPFC  DS    F           PFCOUNT
1 DIVP#    EQU   (*-DIVPPA)  LENGTH OF STRUCTURE
```

Beispiele

Beispiel 1: Lesen und Aktualisieren einer existierenden Datei

```
*-------------------------------------------------------------------*
* Beispiel 1: Lesen und Aktualisieren einer existierenden Datei.    *
*-------------------------------------------------------------------*
D1       DIV     MF=D
*
BSP001   START
         BALR  R10,0
         USING *,R10
         USING D1,R9
         LA    R9,PA
*-------------------------------------------------------------------*
* Eröffnen der Datei mit INOUT                                      *
*-------------------------------------------------------------------*
         DIV   MF=E,PARAM=PA
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*-------------------------------------------------------------------*
* Ermitteln der Dateigröße und Anfordern der Seiten für ein Fenster *
* REQM liefert die Anfangsadresse in R1.                            *
*-------------------------------------------------------------------*
         L     R3,DIVPSIZE        DATEIGROESSE IN 4KB-SEITEN -> R3
         REQM  (R3),PARMOD=31
         LTR   R15,R15
         BNZ   ERROR
         LR    R8,R1
*-------------------------------------------------------------------*
* Erzeugen eines Fensters, in dem die Seiten der Datei erst bei einem *
* Zugriff erscheinen (DISPOS=*OBJECT und LOCVIEW=*NONE sind Standard) *
* Das Fenster wird in der Dateigröße angelegt (OFFSET und SPAN sind *
* nicht spezifiziert).                                              *
*-------------------------------------------------------------------*
         DIV   MF=M,PARAM=PA,FCT=*MAP,AREA=(R8)
         DIV   MF=E,PARAM=PA
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*-------------------------------------------------------------------*
* Schreiben geänderter Fensterseiten in die Datei.                  *
*-------------------------------------------------------------------*
         DIV   MF=M,PARAM=PA,FCT=*SAVE
         DIV   MF=E,PARAM=PA
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*-------------------------------------------------------------------*
* Abbau des Fensters und Schließen der Datei.                       *
```

```
* Nach dem Abbau des Fensters sind die Seiten im Zustand wie unmittel-*
* bar nach REQM.                                                      *
*--------------------------------------------------------------------*
*
         DIV   MF=M,PARAM=PA,FCT=*UNMAP,DISPOS=*FRESH
         DIV   MF=E,PARAM=PA
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*
         DIV   MF=M,PARAM=PA,FCT=*CLOSE
         DIV   MF=E,PARAM=PA
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*
*
ERROR    DS   0Y
*
*
PA       DIV   MF=L,FCT=*OPEN,LINK='TST001',MODE=*INOUT
         END
```

Beispiel 2: Kopieren und Modifizieren einer Datei

```
*--------------------------------------------------------------------*
* Beispiel 2: Kopieren und Modifizieren einer Datei.                  *
*--------------------------------------------------------------------*
*
D1       DIV     MF=D
*
BSP002   START
         BALR  R10,0
         USING *,R10
         USING D1,R9
         LA    R9,PA1
*--------------------------------------------------------------------*
* Eröffnen einer existierenden Datei.                                 *
*--------------------------------------------------------------------*
         DIV   MF=E,PARAM=PA1
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*--------------------------------------------------------------------*
* Ermitteln der Dateigröße und Anfordern der Seiten für ein Fenster   *
* REQM liefert die Anfangsadresse in R1.                              *
*--------------------------------------------------------------------*
         L     R3,DIVPSIZE        DATEIGROESSE IN 4KB-SEITEN -> R3
         REQM  (R3),PARMOD=31
         LTR   R15,R15
         BNZ   ERROR
```

```
         LR    R8,R1
*-------------------------------------------------------------------------*
* Erzeugen eines Fensters, wobei die Seiten der Datei sofort einge-       *
* lesen werden (Die Seiten der Datei erscheinen im Fenster wegen          *
* DISPOS=*OBJECT (Standardfall), die Seiten werden sofort eingelesen      *
* wegen LOCVIEW=*MAP).                                                    *
* Das Fenster wird in der Dateigröße angelegt (OFFSET und SPAN sind       *
* nicht spezifiziert).                                                    *
*-------------------------------------------------------------------------*
         DIV   MF=M,PARAM=PA1,FCT=*MAP,AREA=(R8)
         DIV   MF=E,PARAM=PA1
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*-------------------------------------------------------------------------*
* Modifizieren von Fensterseiten.                                         *
*-------------------------------------------------------------------------*
*
*
*-------------------------------------------------------------------------*
* Abbau des Fensters und Schließen der Datei.                             *
* Der Inhalt der Fensterseiten bleibt erhalten (wegen DISPOS=*UNCHNG).*
*-------------------------------------------------------------------------*
*
         DIV   MF=M,PARAM=PA1,FCT=*UNMAP,DISPOS=*UNCHNG
         DIV   MF=E,PARAM=PA1
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*
         DIV   MF=M,PARAM=PA1,FCT=*CLOSE
         DIV   MF=E,PARAM=PA1
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*
*-------------------------------------------------------------------------*
* Öffnen der neuen Datei.                                                 *
*-------------------------------------------------------------------------*
         LA    R9,PA2
         DIV   MF=E,PARAM=PA2
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*-------------------------------------------------------------------------*
* Erzeugen eines Fensters, wobei die Daten erhalten bleiben (wegen        *
* DISPOS=*UNCHNG).                                                        *
* Die Bereichsadresse ist noch in R8, die Fenstergröße in R3              *
*-------------------------------------------------------------------------*
         DIV   MF=M,PARAM=PA2,FCT=*MAP,AREA=(R8),DISPOS=*UNCHNG,SPAN=(R3)
         DIV   MF=E,PARAM=PA2
         CLI   DIVPSR1,DIVPRFSP
```

```
         BNE   ERROR
*-------------------------------------------------------------------*
* Schreiben der Fensterseiten in die neue Datei.                     *
*-------------------------------------------------------------------*
         DIV   MF=M,PARAM=PA2,FCT=*SAVE
         DIV   MF=E,PARAM=PA2
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*-------------------------------------------------------------------*
* Abbau des Fensters und Schliessen der Datei.                       *
*-------------------------------------------------------------------*
         DIV   MF=M,PARAM=PA2,FCT=*UNMAP,DISPOS=*FRESH
         DIV   MF=E,PARAM=PA2
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*
         DIV   MF=M,PARAM=PA2,FCT=*CLOSE
         DIV   MF=E,PARAM=PA2
         CLI   DIVPSR1,DIVPRFSP
         BNE   ERROR
*
*
ERROR    DS   0Y
*
*
PA1      DIV   MF=L,FCT=*OPEN,LINK='TST001',LOCVIEW=*MAP
PA2      DIV   MF=L,FCT=*OPEN,LINK='TST002',MODE=*OUTIN
         END
```

DROPTFT – TFT-Eintrag freigeben

Makrotyp: S-Typ (C-Form/D-Form/E-Form/L-Form/M-Form) (siehe Seite 870)

Der DROPTFT-Makroaufruf gibt eine LOCK-FILE-LINK-Sperre für einen Eintrag in der Task File Table (TFT) frei. Steht für diesen Eintrag noch ein REMOVE-FILE-LINK-Kommando bzw. ein RELTFT-Makroaufruf an, wird dieser jetzt bearbeitet , d.h. der TFT-Eintrag wird entsprechend den Angaben im Kommando/Makro gelöscht, und die mit ihm verbundenen privaten Geräte werden freigegeben.

Format

Operation	Operanden
DROPTFT	,LINK = <c-string 1..8> / <var: char:8> ,VERSION = <integer 1..1> ,MF = C / D / E / L / M ,PARAM = <adr> / <(r)> ,PREFIX = D / <pre> ,MACID = MAD / <macid>

Operandenbeschreibung

LINK
Dateikettungsname (Linkname) des freizugebenden TFT-Eintrags.

Voreinstellung: Der erste TFT-Eintrag mit dem Linknamen *BLANK wird freigegeben.

= <c-string 1..8>
Dateikettungsname (Angabe in Hochkommas)

= <var: char: 8>
Name einer Variablen, die den Dateikettungsnamen enthält

MACID
wird nur in Verbindung mit MF=C/D/M ausgewertet und legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: MACID = MAD

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

Voreinstellung: Operandenliste und SVC wie bisher

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

= adr
adr ist die symbolische Adresse (der Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

PREFIX
wird nur in Verbindung mit MF=C/D/M ausgewertet und legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein Zeichen langes Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen sollen.

VERSION = <integer 1..1>
Kontrolloperand; steuert Generierung

Hinweis zur Programmierung

Der Fehlercode wird im Standardheader des Parameterbereichs zurückgeliefert. Eine Programm-Terminierung mit STXIT-Anschluss kann in folgenden Fällen eingeleitet werden:

– Parameteradresse fehlerhaft (z.B. kürzer als der Standardheader)

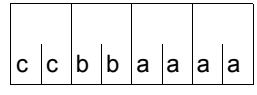
– Parameteradresse nicht wort-ausgerichtet

– UNIT oder FUNCTION im Header fehlerhaft

– Header nicht beschreibbar

Returncodes

Standard-header:



Über die Ausführung des Makros DROPTFT wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'02'	X'00'	X'0662'	Linkname fehlt oder ungültig. Keine Aktion nötig
	X'40'	X'06FF'	BCAM-Verbindung nicht betriebsbereit oder wird geschlossen
	X'01'	X'xxxx'	RFA für Version < 12 nicht unterstützt

EAM – EAM-Dateien verarbeiten

Makrotyp: R-Typ

Alle die Zugriffsmethode EAM betreffenden Anforderungen werden über den EAM-Makroaufruf abgewickelt. Die auszuführende Operation ist durch den Inhalt des Mini-FCB festgelegt.

Format

Operation	Operanden
EAM	$\left\{\begin{array}{l}\texttt{mfcbadr}\\\texttt{(1)}\end{array}\right\}$[,PARMOD=$\left\{\begin{array}{l}\texttt{24}\\\texttt{31}\end{array}\right\}$]

Operandenbeschreibung

mfcbadr
Adresse des Mini-FCB

(1)
Die Adresse des Mini-FCB steht im Register 1.

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.

Voreinstellung:
der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst.
Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

Hinweis zur Programmierung

Der EAM-Makroaufruf zerstört die Register 0, 1, 14 und 15.

Returncodes

bei PARMOD=24 in Register 15; bei PARMOD=31 im Feld IDMRETCO

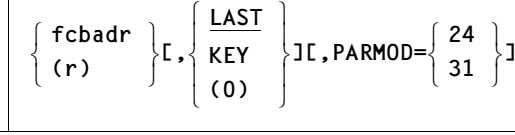
Returncode	Bedeutung
0	Operation erfolgreich abgeschlossen
4	Operation nicht erfolgreich abgeschlossen: Fehlerbyte überprüfen
8	geprüfte Ein-/Ausgabeoperation nicht beendet (nach Prüfoperationen)

ELIM – Satz streichen

Makrotyp: R bei PARMOD=24
0 bei PARMOD=31

Der ELIM-Makroaufruf streicht einen Satz aus einer ISAM-Datei. Der zweite Operand (LAST / KEY / (0)) gibt an, welcher Satz zu streichen ist.

Format

Operation	Operanden
ELIM	

Operandenbeschreibung

fcbadr
Adresse des FCB der zu verarbeitenden Datei

(1)
Die FCB-Adresse steht im Register 1.

LAST
Es wird der zuletzt von einem der Makroaufrufe GET, GETFL, GETKY oder GETR bereitgestellte Satz bearbeitet (Voreinstellung).

Zwischen den Makroaufrufen GET, GETR, GETFL oder GETKY und dem zugehörigen Makroaufruf ELIM mit der Funktion LAST darf kein anderer ISAM-Makroaufruf ausgeführt werden, der sich auf denselben FCB bezieht (Ausnahme: OSTAT). Bei SHARUPD=YES muss außerdem mit dem vorangehenden Leseaufruf eine Sperre gesetzt werden (LOCK), die beim ELIM-Makroaufruf noch bestehen muss. D.h.: es darf auch auf einen anderen FCB keine Aktion ausgeführt werden, die zur Aufhebung der Sperre führt.

KEY
Der Schlüssel des zu bearbeitenden Satzes steht an der Adresse, die im FCB mit KEYARG definiert wurde.

Ist der angegebene Schlüssel nicht vorhanden, wird das das Anwenderprogramm an der NOFIND-Adresse fortgesetzt (siehe Seite 401). Existieren in einer Datei Sätze mit gleichem Schlüssel, wird der erste Satz dieser Gruppe gestrichen.

(0)
Der Inhalt des Registers 0 zeigt an, welcher Satz bearbeitet werden soll:
– Ist die Adresse im Register 0 nicht die des FCB, wird die Funktion „KEY“ ausgelöst.
– Ist die Adresse im Register 0 die Adresse des FCB, wird die Funktion „LAST“ ausgelöst, d.h. der zuletzt gelesene Satz wird bearbeitet.

PARMOD
gibt den Generierungsmodus an.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24
Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (nur 24-Bit-Adressierung).

= 31
Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist (24-Bit- oder 31-Bit-Adressierung).

Hinweis zur Programmierung

Der ELIM-Makroaufruf zerstört die Register 0, 1, 14 und 15.

ENCFILE – Unverschlüsselte Datei in verschlüsselte Datei umwandeln

Makrotyp: S-Typ (E-Form/M-Form/L-Form/C-Form/D-Form) (siehe Seite 870)

Der ENCFILE-Makroaufruf wandelt eine unverschlüsselte Datei in eine verschlüsselte Datei um.

Durch die Dateiverschlüsselung mit Crypto-Kennwort ist es möglich, den Inhalt einer Datei vor unbefugtem Zugriff zu schützen – auch gegenüber Personen mit TSOS-Privileg. Dateiverschlüsselung beinhaltet aber keinen Schutz gegen Löschen, Überschreiben oder Zerstören des Dateiinhalts und kann Dateischutz (z.B. durch ein Schreibkennwort) nicht ersetzen.

Das verwendete Verschlüsselungsverfahren wird dem Klasse-2-Systemparameter FILECRYP entnommen.

Nach ENCFILE sind die Verschlüsselungsmerkmale (Verfahren und Kontrollstring für Crypto-Kennwort) im Katalogeintrag eingetragen sowie das Lese- und Ausführungskennwort gelöscht.

Es können nur Plattendateien auf Pubsets verschlüsselt werden.

Beim Verschlüsseln von PAM-Dateien wird der Last Byte Pointer auf Blockgrenze hochgesetzt.

Dateigenerationen

ENCFILE kann nicht für einzelne Dateigenerationen angewendet werden, sondern nur für komplette Dateigenerationsgruppen. Innerhalb einer Dateigenerationsgruppe haben alle Generationen mit der Ausnahme von Bandgenerationen die gleichen Verschlüsselungsmerkmale wie der Gruppeneintrag.

Format

Operation	Operanden
ENCFILE	,PATHNAM=<c-string 1..54: filename 1..54> / <var: char:54> .CRYPASS=<c-string 1..8: filename 1..8> / <var: char:8> ,REFFILE=<c-string 1..54: filename 1..54> / <var: char:54> ,EQUATES=YES / NO MF=L
	MF=D,PREFIX=D / <pre>
	MF=E,PARAM=<name 1..27>
	MF=C / M ,PREFIX=D / <pre> ,MACID=MAE / <macid>

Operandenbeschreibung

PATHNAM
Gibt die Datei an, die verschlüsselt werden soll. Das Crypto-Kennwort der Datei muss in der Crypto-Kennworttabelle der aufrufenden Task stehen.

Die zu verschlüsselnde Datei muss folgende Voraussetzungen erfüllen:
– Sie muss unverschlüsselt sein.
– Sie muss bereits einen Katalogeintrag haben.
– Der Pubset, auf dem sie katalogisiert ist, muss lokal verfügbar sein.
– Sie darf nicht auf einer Privatplatte liegen.
– Sie darf keinen Bandtyp haben.
– Sie darf nicht unter der Benutzerkennung TSOS auf dem Home-Pubset liegen.

=<c-string 1..54: filename 1..54>
Vollqualifizierter Pfadname der Datei.

=<var: char:54>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 54 Byte, in dem der Pfadname der Datei abgelegt ist.

CRYPASS
Der Operand darf nicht zusammen mit dem Operanden REFFILE angegeben werden.
Explizite Angabe des Crypto-Kennworts, das für alle Zugriffe auf den entschlüsselten Dateiinhalt gebraucht wird. Anstelle dieser Angabe kann des Cypto-Kennwort auch von einer verschlüsselten Referenzdatei übernommen werden (siehe Operand REFFILE).

Wenn im Klasse-2-Systemparameter FREFCRYP eine Benutzerkennung eingetragen ist, so darf CRYPASS nur dann angegeben werden, wenn die unter PATHNAM angegebene Datei auf dieser Benutzerkennung liegt.

=<c-string 1..8: filename 1..8>
Crypto-Kennwort.

=<var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 54 Byte, in dem das Crypto-Kennwort abgelegt ist.

REFFILE
Der Operand darf nicht zusammen mit dem Operanden CRYPASS angegeben werden.
Verschlüsselte Referenzdatei, von der das Crypto-Kennwort übernommen wird.

Das Crypto-Kennwort der Referenzdatei muss in der Crypto-Kennworttabelle der aufrufenden Task eingetragen sein.

=<c-string 1..54: filename 1..54>
Vollqualifizierter Pfadname der Referenzdatei.

=<var: char:54>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 54 Byte, in dem der Pfadname der Referenzdatei abgelegt ist.

EQUATES
Gibt an, ob bei der Expansion des Parameterbereichs auch Equates für die Werte der Felder des Parameterbereichs generiert werden sollen.

= YES
Bei der Expansion des Parameterbereichs werden auch Equates für die Werte der Felder des Parameterbereichs generiert.

= NO
Bei der Expansion des Parameterbereichs werden keine Equates für die Werte der Felder des Parameterbereichs generiert.

Beispiel

```
:
MVC ENCFMFC(XMAE#),ENCFMFL
ENCFILE MF=M,PREFIX=X,PATHNAM='UMSATZ.3.QUARTAL.2004'
ENCFILE MF=E,PARAM=ENCFMFC
:
ENCFMFC ENCFILE MF=C,PREFIX=X
ENCFMFL ENCFILE MF=L,CRYPASS='KROKODIL'
:
```

Hinweise zur Programmierung

1. Vor der Generierung des Layouts des Parameterbereichs muss der Standard-Header aufgelöst werden.
2. Alle RESERVED-Felder des Parameterbereichs müssen mit binären Nullen gelöscht sein.
3. Bei allen Änderungen im Parameterbereich, die nicht mit Hilfe von GCs vorgenommen werden, ist der Aufrufer selbst für die Konsistenz des Parameterbereichs verantwortlich.
4. Beim nichtprivilegierten Aufruf (Funktionszustand TU) zeigt Register 1 auf den Parameterbereich.
5. Für die im Parameterbereich stehenden Namen erfolgt während der Funktionsausführung keine Umwandlung von Klein- in Großbuchstaben. Dagegen kann bei der GC-Expansion je nach Compiler-Einstellung eine Umwandlung von Klein- in Großbuchstaben erfolgen.

Hinweise zur Funktionsausführung

– Dateisperren und Dateischutzattribute, die den Schreibzugriff auf den Katalogeintrag oder auf den Inhalt einer Datei verbieten, verhindern damit auch das Umwandeln der Datei mit ENCFILE.

– Das Umwandeln einer Datei mit ENCFILE erfordert das Eigentumsrecht der aufrufenden Task an der Datei. Die Umwandlung erfolgt also, wenn:

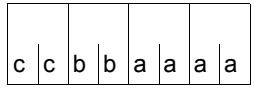
  – Die Datei unter der Benutzerkennung der aufrufenden Task liegt.
  – Die aufrufende Task unter einer Benutzerkennung mit dem Privileg TSOS läuft.
  – Die Benutzerkennung der aufrufenden Task Mit-Eigentümer der Datei und die Datei nicht temporär ist.

- Die Umwandlung der verschlüsselten Datei wird mit SAT protokolliert.
  Das hierbei ausgegebene AUDIT-Attribut wird dem Katalogeintrag der umzuwandelnden Datei entnommen (siehe Kommando CREATE-FILE, Operand AUDIT, im Handbuch „Kommandos“ [3]).
- Zusätzliche Funktionen für Tasks mit Privileg TSOS:
  Wenn die aufrufende Task das Privileg TSOS hat, sind zusätzlich folgende Funktionen möglich:
  - Es können auch temporäre Dateien, die nicht zur aufrufenden Task, sondern zu einer anderen Task gehören, angegeben werden.
  - Es können auch temporäre Dateien auf einem anderen Pubset als dem Default-Pubset der Benutzerkennung angelegt werden. (Diese werden bei Beendigung der aufrufenden Task nicht automatisch gelöscht.)
- RFA:
  ENCFILE wird abgewiesen, wenn auf die umzuwandelnde Datei nur über RFA zugegriffen werden kann.
- Hilfsdatei:
  Beim Umwandeln mit ENCFILE wird eine Hilfsdatei angelegt und bei Beendigung der Funktionsausführung automatisch gelöscht. In die Hilfsdatei wird der umgewandelte Dateiinhalt geschrieben. Die Hilfsdatei benötigt ebenso viel Plattenspeicherplatz wie die umzuwandelnde Datei.
  Der Dateiname der Hilfsdatei hat folgenden Aufbau:
  `S.DMS.<tsn>.<date><time>.CRYPTO`

Returncodes

Der Returncode wird im Standardheader der Parameterliste zurückgeliefert. Der Standardheader darf nicht im Read-only-Bereich liegen, sonst erfolgt Programmterminierung.

Standard-header:



Über die Ausführung des Makros ENCFILE wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	**X'bb'**	**X'aaaa'**	**Erläuterung**
X'00'	X'00'	X'0000'	Kein Fehler
	X'01'	X'0554'	Format des Dateinamens unzulässig
	X'01'	X'0576'	a) Fehlerhafte Operandenkombination b) Nicht gelöschte UNUSED-Felder
	X'20'	X'0578'	Interner Fehler bei Überprüfung der Zugriffsrechte
	X'82'	X'0594'	Nicht genügend virtueller Speicher verfügbar
	X'20'	X'05C7'	Interner Fehler im DVS
	X'01'	X'05CB'	Fehlerhafter/unerlaubter erster Dateiname
	X'40'	X'05CF'	Kennwort nicht in Kennworttabelle
	X'20'	X'05EC'	Interner Fehler bei Crypto-Kennwort-Behandlung
	X'40'	X'05FD'	Datei ist schreibgeschützt
	X'40'	X'0609'	Aktion für Systemdatei nicht erlaubt
	X'40'	X'060D'	Fehlerhafter Dateiname für Referenzdatei angegeben
	X'40'	X'0663'	Verschlüsselung der Datei nicht erlaubt
	X'40'	X'0666'	Dateischutz verhindert Zugriff
	X'40'	X'0667'	Datei nicht als Referenzdatei verwendbar
X'02'	X'00'	X'0669'	Schutzmerkmal implizit geändert
X'00'	X'40'	X'066A'	Crypto-Kennwort kann nicht verwendet werden
	X'40'	X'066D'	Crypto-Kennwort-Angabe wurde eingeschränkt
	X'00'	X'066E'	Hilfsdatei verwenden
	X'01'	X'06C8'	Attribut unzulässig für Dateigeneration
	X'01'	X'FFFF'	Falsche Funktionsnummer im Standardheader
	X'03'	X'FFFF'	Falsche Versionsnummer im Standardheader

ERASE – Dateien löschen

Makrotyp: S-Typ (E-Form/L-Form/D-Form) (siehe Seite 870)

Mit dem ERASE-Makroaufruf kann der Anwender eigene temporäre oder permanente Dateien, Dateigenerationsgruppen oder Dateigenerationen löschen, abhängig von Auswahlkriterien, die im Makroaufruf anzugeben sind. Außerdem kann der Anwender Speicherplatz freigeben und Dateien exportieren (= Katalogeintrag löschen). Die Operanden des ERASE-Makroaufrufs lassen sich in fünf Gruppen einteilen, die den verschiedenen Funktionsebenen entsprechen (siehe Tabelle „Funktionsübersicht“ auf Seite 324).

Selektion

Mit Selektionsoperanden definiert der Anwender, welche Dateien/Katalogeinträge bearbeitet werden. Als Auswahlkriterien dienen im Katalogeintrag hinterlegte Eigenschaften.
Zu diesem Zweck sind im ERASE-Makroaufruf Operanden des FSTAT integriert.
Wird ein Selektionsoperand nicht angegeben, ist die Auswahl der zu bearbeitenden Dateien/Katalogeinträge unabhängig von dem Auswahlkriterium.

Dateischutz

Dateischutzoperanden ermöglichen es dem Anwender, Dateien zu löschen, für die Schutzmerkmale wie Zugriffsart, Kennwörter u.a. definiert wurden, auch wenn diese Schutzmerkmale zuvor nicht zurückgesetzt wurden.

Makroausführung

Aktionsoperanden steuern den internen Ablauf der ERASE-Bearbeitung. Der Anwender kann zum einen den Umfang des Löschens bestimmen, zum anderen aber auch Bedingungen für das Löschen definieren.

Kontrolloperanden erlauben es dem Anwender, sich die Benutzerschnittstelle in gewissem Rahmen selbst zu definieren. Er kann z.B. ein Protokoll ausgeben lassen oder im Dialog Kontrollfragen beantworten.

Makrogenerierung

Übersetzeroperanden steuern die Makrogenerierung, der Operand VERSION z.B. steuert die Generierung der Operandenliste.

Für bestehende Programme wird Source-Kompatibilität gewährleistet, da das neue Format VERSION=3 die Funktionen der alten Formate VERSION=0/1/2 voll abdeckt. Wenn im Programm die generierte Operandenliste verändert wird, muss das Programm mit dem geänderten Makroaufruf neu übersetzt werden.

ACL-geschützte Dateien

Ist eine Datei durch eine ACL geschützt, kann sie vom Dateieigentümer oder dem Systemverwalter (Kennung mit dem Privileg TSOS) nur unter Angabe des Operanden IGNORE=*ACCESS gelöscht werden.

Seit SECOS V4.0 wird die Zugriffskontrolle über ACL nicht mehr unterstützt. ACL-geschützte Dateien sollten deshalb mit GUARDS geschützt werden. Der Schutz durch GUARDS „überschreibt“ den ACL-Schutz und macht die Datei wieder zugänglich für den Dateieigentümer (und den Personenkreis, dem der Dateieigentümer den Zugriff erlaubt).

Funktionsübersicht

Operand	Operandenwert	Auswahlkriterium
Selektion – Dateiname		
*DUMMY		Pseudodatei (DUMMY-Datei)
pfadname		Pfadname (Angabe voll-, teilqualifiziert, Muster)
prefix		temporäre Anwenderdateien
*SYSid		Systemdateien, Muster erlaubt
*		EAM-Objektmoduldatei
Selektion – Dateityp		
BLKCTRL	NONE PAMKEY/DATA/DATA2K/ DATA4K/NO/NK/NK2/NK4	Katalogeinträge nicht eröffneter Dateien Dateiformat
FCBTYPE	NONE ISAM/SAM/BTAM/PAM	Katalogeinträge nicht eröffneter Dateien Zugriffsmethode
FILTYPE	*ANY/*BS2000/*NODE	Dateityp auf Net-Storage: BS2000-Datei oder Node-File
POS	AFTER/BEFORE	in Verbindung mit TYPE=FGG; bestimmt die zu bearbeitenden Dateigenerationen
TYPE	FILE FGG PLAM	Dateien, nicht FGG oder Dateigenerationen Dateigenerationen oder FGG PLAM-Bibliotheken
WORKFIL	*NO/*YES	Arbeitsdateien

(Teil 1 von 5)

Operand	Operandenwert	Auswahlkriterium
Selektion – Datenträger		
STOTYPE	*PUBSPACE/*NETSTOR	Speichertyp
SUPPORT	PUBLIC PRDISC TAPE	Dateien auf Public Disk Dateien auf Privatplatte Band
VOLSET		Volume-Set
VOLUME		Archivnummer des Datenträgers
Selektion – Datensicherheit/ Datenschutz		
ACCESS	READ/WRITE	Schreibschutz
ACL	YES/NO	Schutz mit ACL
AVAIL	*STD/*HIGH	Ausfallsicherheit
BACKUP	A/B/C/D/E	BACKUP-LEVEL
BASACL	NONE/YES	Schutz mit BASIC-ACL
ENCRYPT	ANY/NONE/AES/DES	verschlüsselte Dateien
GROUPAR	NO-ACCESS/zugriffsliste	Zugriffsrechte der Benutzergruppe
GUARDS	(READ..., WRITE...,EXEC...)	Schutz mit GUARDS
OTHERAR	NO-ACCESS/zugriffsliste	Zugriffsrechte der anderen Benutzer
OWNERAR	NO-ACCESS/zugriffsliste	Zugriffsrechte des Eigentümers
PASS	NONE EXPASS RDPASS WRPASS	Kennwortschutz
PROTACT	ANY/LEVEL-0/ LEVEL-1/LEVEL-2	Schutzstufe der aktivierten Zugriffskontrolle
SHARE	NO/YES/SPECIAL	Mehrbenutzbarkeit
Selektion – Speicherplatz (Plattendateien)		
EXTENTS		Anzahl der Extents
FSIZE		Größe des reservierten, aber nicht belegten Speicherplatzes
LASTPAG		Anzahl der belegten PAM-Seiten
RELSPAC		Sperre gegen Freigabe von Speicherplatz
SIZE		Größe des reservierten Speicherplatzes

(Teil 2 von 5)

Operand	Operandenwert	Auswahlkriterium
Selektion – Speicherplatz (Banddateien)		
BLKCNT		Anzahl der Blöcke der Datei auf Band
Selektion – Datums- und Zeitangabe		
CRDATE		Erstellungsdatum und -zeit
DELDATE		DELETION-Datum und -Zeit (implizit: Schutzfrist)
EXDATE		Freigabedatum und -zeit (implizit: Schutzfrist)
LADATE		Datum und Zeit des letzten Zugriffs
LCDATE		Datum und Zeit des letzten schreibenden Zugriffs
TIMBASE	*UTC/*LTI	Zeitbasis der Datumsangaben
Selektion – HSMS		
MIGRATE	ALLOWED INHIBIT FORBIDDEN	Migration zugelassen/ Migration kurzfristig zugelassen/ Migration nicht zugelassen
SLEVEL	S0/S1/S2	Speicherhierarchie-Ebene
S0MIGR	*ALLOWED/*FORBIDDEN	Migrations-Erlaubnis
Selektion – Performance / Ein-/Ausgabe-Attribute		
DISKWR	IMMEDIATE/BY-CLOSE	Zeitpunkt des Zurückschreibens auf Platte
IOPERF	STD/HIGH/VERY-HIGH	Performance-Attribut
IOUSAGE	RDWRT/WRITE/READ	Art der Ein/Ausgabe-Operation
Selektion – Dateizustand		
STATE	NOCLOS CLOSED CACHED NOT-CACHED CACHE-NOT-SAVED REPAIR-NEEDED DEFECT-REPORTED OPEN-ALLOWED NO-OPEN-ALLOWED	aktueller Zustand der Datei
Selektion – Codierung		
CCS		Codiertabelle (Coded Character Set)
Selektion – Metainformation		
ADMINFO	*NONE/<c-string 1..8>	Systemverwalter-Metainformation
USRINFO	*NONE/<c-string 1..8>	Benutzer-Metainformation

(Teil 3 von 5)

Operand	Operandenwert	Auswahlkriterium
Selektion – SM-Pubset		
MANCLAS	*NONE/ <c-string 1..8>	Management-Klasse
PREFORM		beabsichtigtes Dateiformat auf SM-Pubsets
STOCLAS	*NONE/ <c-string 1..8>	Storage-Klasse
Dateischutz – Dateischutzoperanden		
IGNORE	keine Angabe	vereinbarte Schutzmerkmale werden berücksichtigt
	ACCESS	der Zugriffsschutz über ACCESS=READ, BASIC-ACL, ACL und GUARDS wird ignoriert
	EXDATE	Schutzfristen werden ignoriert
	RDPASS WRPASS EXPASS	*nur für Systemverwaltung:* das vereinbarte Kennwort wird ignoriert
PASSWORD	keine Angabe kennwort	Kennwortschutz wird berücksichtigt Kennwortschutz, der durch das angegebene Kennwort definiert wurde, wird ignoriert
Makroausführung – Aktionsoperanden		
CATALOG		Dateien auf privaten Datenträgern werden exportiert
DATA		logisches Löschen: die datenbezogenen Eigenschaften der Datei werden gelöscht, der Katalogeintrag entsprechend geändert, die Speicherplatzzuweisung bleibt erhalten
DATA-KEEP-ATTR		logisches Löschen wie bei DATA, aber die datenbezogenen Eigenschaften bleiben erhalten
DELETE-OR-EXPORT		Dateien auf privaten Datenträgern und Node-Files auf Net-Storage-Volumes werden exportiert, Dateien auf Public-Platten oder auf Net-Storage-Volumes werden gelöscht
DESTROY		der Katalogeintrag wird gelöscht, der Speicherplatz freigegeben und überschrieben
MOUNT		Dateien auf Privatplatten: legt fest, ob alle betroffenen Platten on-line sein müssen
SPACE		nur Speicherplatzfreigabe, der Katalogeintrag bleibt erhalten

(Teil 4 von 5)

Operand	Operandenwert	Auswahlkriterium
SPACE-CATALOG		der Katalogeintrag wird gelöscht, Speicherplatz freigegeben
Makroausführung – Kontrolloperanden		
CHECK	NO	kein Eingriff durch den Anwender möglich (Voreinstellung für Prozeduren und Batchbetrieb)
	MULTIPLE	Dialog bei Wechsel von Katalog- oder Benutzerkennung, wenn „pfadname“ nicht vollqualifiziert angegeben wurde (Voreinstellung im Dialogbetrieb)
	ERROR	Dialog bei Auftreten eines vom Aufrufer behebbaren Fehlers
	PVS	Dialog bei Wechsel der Katalogkennung
	SINGLE	der Anwender bestimmt für jede ausgewählte Datei im Dialog, ob sie vom aktuellen ERASE-Makroaufruf bearbeitet werden soll
	USERID	*nur für Systemverwaltung:* Dialog bei Wechsel der Benutzerkennung
LIST	NO/YES	Löschvorgang [nicht] auf SYSOUT protokollieren
NOSTEP	errcode	der Anwender kann über den DVS-Fehlerschlüssel definieren, welche Fehler keinen Spin-off auslösen
Makrogenerierung – Übersetzeroperanden		
MF		Makrogenerierung (Operandenliste/SVC/DSECT)
PREFIX	prefix	aufrufspezifisches Präfix
VERSION	0	Makroformat BS2000-Version < V9.5A (s. Tabelle „Versionsunterschiede VERSION=0/1/2“ auf Seite 390)
	1	Makroformat BS2000-Version V9.5A und V10.0A (s. Tabelle „Versionsunterschiede VERSION=0/1/2“ auf Seite 390)
	2	Makroformat ab Version BS2000/OSD-BC V1.0 (siehe Format und Operandenbeschreibung)
	3	Makroformat ab Version BS2000/OSD-BC V3.0 (siehe Format und Operandenbeschreibung)

(Teil 5 von 5)

Operandenübersicht

Operand	Operandenwert	Funktion
*DUMMY		Selektionsoperand – Pseudodatei (DUMMY-Datei)
pfadname		Selektionsoperand – Pfadname (Angabe voll-, teilqualifiziert, Muster)
prefix		Selektionsoperand – temporäre Anwenderdateien
*SYSid		Selektionsoperand – Systemdateien, Muster erlaubt
*		Selektionsoperand – EAM-Objektmoduldatei
CATALOG		Dateien auf privaten Datenträgern werden exportiert
DATA		logisches Löschen: die datenbezogenen Eigenschaften der Datei werden gelöscht, der Katalogeintrag entsprechend geändert, die Speicherplatzzuweisung bleibt erhalten
DATA-KEEP-ATTR		logisches Löschen wie bei DATA, aber die datenbezogenen Eigenschaften bleiben erhalten
DELETE-OR-EXPORT		Dateien auf privaten Datenträgern und Node-Files auf Net-Storage-Volumes werden exportiert, Dateien auf Public-Platten oder auf Net-Storage-Volumes werden gelöscht
DESTROY		der Katalogeintrag wird gelöscht, der Speicherplatz freigegeben und überschrieben
MOUNT		Dateien auf Privatplatten: legt fest, ob alle betroffenen Platten on-line sein müssen
SPACE		nur Speicherplatzfreigabe, der Katalogeintrag bleibt erhalten
SPACE-CATALOG		der Katalogeintrag wird gelöscht, Speicherplatz freigegeben
ACCCNT		Selektionsoperand – Zugriffszähler
ACCESS	READ/WRITE	Selektionsoperand – Schreibschutz
ACL	YES/NO	Selektionsoperand – Schutz mit ACL
ADMINFO	*NONE/ <c-string 1..8>	Selektionsoperand – Systemverwalter-Metainformation
AVAIL	*STD/*HIGH	Selektionsoperand – Ausfallsicherheit
BACKUP	A/B/C/D/E	Selektionsoperand – BACKUP-LEVEL
BASACL	NONE/YES	Selektionsoperand – Schutz mit BASIC-ACL
BLKCNT		Selektionsoperand – Anzahl der Blöcke der Datei auf Band

(Teil 1 von 5)

Operand	Operandenwert	Funktion
BLKCTRL	NONE PAMKEY/DATA/ DATA2K/DATA4K/NO/NK/NK2/NK4	Selektionsoperand – Katalogeinträge nicht eröffneter Dateien Dateiformat
CCS		Selektionsoperand – Codiertabelle (Coded Character Set)
CHECK	NO MULTIPLE ERROR PVS SINGLE USERID	kein Eingriff durch den Anwender möglich (Voreinstellung für Prozeduren und Batchbetrieb) Dialog bei Wechsel von Katalog- oder Benutzerkennung, wenn „pfadname“ nicht vollqualifiziert angegeben wurde (Voreinstellung im Dialogbetrieb) Dialog bei Auftreten eines vom Aufrufer behebbaren Fehlers Dialog bei Wechsel der Katalogkennung der Anwender bestimmt für jede ausgewählte Datei im Dialog, ob sie vom aktuellen ERASE-Makroaufruf bearbeitet werden soll *nur für Systemverwaltung:* Dialog bei Wechsel der Benutzerkennung definieren, welche Fehler keinen Spin-off auslösen
CRDATE		Selektionsoperand – Erstellungsdatum und -zeit
DELDATE		Selektionsoperand – DELETION-Datum und -Zeit (implizit: Schutzfrist)
DISKWR	IMMEDIATE/ BY-CLOSE	Selektionsoperand – Zeitpunkt des Zurückschreibens auf Platte
EXDATE		Selektionsoperand – Freigabedatum und -zeit (implizit: Schutzfrist)
EXTENTS		Selektionsoperand – Anzahl der Extents
FCBTYPE	NONE ISAM/SAM/ BTAM/PAM	Selektionsoperand – Katalogeinträge nicht eröffneter Dateien Zugriffsmethode
FILTYPE	*ANY/*BS2000/ *NODE	Selektionsoperand – Dateityp auf Net-Storage (BS2000-Datei oder Node-File)

(Teil 2 von 5)

Operand	Operandenwert	Funktion
FSIZE		Selektionsoperand – Größe des reservierten, aber nicht belegten Speicherplatzes
GROUPAR	NO-ACCESS/ zugriffsliste	Selektionsoperand – Zugriffsrechte der Benutzergruppe
GUARDS	(READ..., WRITE..., EXEC...)	Selektionsoperand – Schutz mit GUARDS
IGNORE	keine Angabe	vereinbarte Schutzmerkmale werden berücksichtigt
	ACCESS	der Zugriffsschutz über ACCESS=READ, BASIC-ACL, ACL und GUARDS wird ignoriert
	EXDATE	Schutzfristen werden ignoriert
	RDPASS WRPASS EXPASS	*nur für Systemverwaltung:* das vereinbarte Kennwort wird ignoriert
IOPERF	STD/HIGH/ VERY-HIGH	Selektionsoperand – Performance-Attribut
IOUSAGE	RDWRT/WRITE/ READ	Selektionsoperand – Art der Ein/Ausgabe-Operation
KEEPACL		*nur für Systemverwalter:* Löschen der Zugriffsliste (ACL)
LADATE		Selektionsoperand – Datum und Zeit des letzten Zugriffs
LASTPAG		Selektionsoperand – Anzahl der belegten PAM-Seiten
LCDATE		Selektionsoperand – Datum und Zeit des letzten schreibenden Zugriffs
LIST	NO/YES	Löschvorgang [nicht] auf SYSOUT protokollieren
MANCLAS	*NONE/ <c-string 1..8>	Selektionsoperand – Management-Klasse
MF		Makrogenerierung (Operandenliste/SVC/DSECT)
MIGRATE	ALLOWED/ INHIBIT/ FORBIDDEN	Selektionsoperand – Migration zugelassen/ kurzfristig zugelassen/ nicht zugelassen
NOSTEP	errcode	der Anwender kann über den DVS-Fehlerschlüssel definieren, welche Fehler keinen Spin-off auslösen
OTHERAR	NO-ACCESS/ zugriffsliste	Selektionsoperand – Zugriffsrechte der anderen Benutzer

(Teil 3 von 5)

Operand	Operandenwert	Funktion
OWNERAR	NO-ACCESS/ zugriffsliste	Selektionsoperand – Zugriffsrechte des Eigentümers
PASS	NONE EXPASS RDPASS WRPASS	Selektionsoperand – Kennwortschutz
PASSWORD	keine Angabe kennwort	Kennwortschutz wird berücksichtigt Kennwortschutz, der durch das angegebene Kennwort definiert wurde, wird ignoriert
POS	AFTER/BEFORE	Selektionsoperand – in Verbindung mit TYPE=FGG; bestimmt die zu bearbeitenden Dateigenerationen
PREFIX	prefix	aufrufspezifisches Präfix
PREFORM		Selektionsoperand – beabsichtigtes Dateiformat auf SM-Pubsets
PROTACT	ANY/LEVEL-0/ LEVEL-1/LEVEL-2	Selektionsoperand – Schutzstufe der aktivierten Zugriffskontrolle
RELSPAC		Selektionsoperand – Sperre gegen Freigabe von Speicherplatz
SHARE	NO/YES/SPECIAL	Selektionsoperand – Mehrbenutzbarkeit
SIZE		Selektionsoperand – Größe des reservierten Speicherplatzes
SLEVEL	S0/S1/S2	Selektionsoperand – Speicherhierarchie-Ebene
STATE	NOCLOS CLOSED CACHED NOT-CACHED CACHE-NOT -SAVED REPAIR-NEEDED DEFECT -REPORTED OPEN-ALLOWED NO-OPEN -ALLOWED	Selektionsoperand – aktueller Zustand der Datei
STOCLAS	*NONE/ <c-string 1..8>	Selektionsoperand – Storage-Klasse
STOTYPE	*PUB- SPACE/*NETSTOR	Selektionsoperand – Speichertyp

(Teil 4 von 5)

Operand	Operandenwert	Funktion
SUPPORT	PUBLIC	Selektionsoperand – Dateien auf Public Disk
	PRDISC	Dateien auf Privatplatte
	TAPE	Datenträger – Band
S0MIGR	*ALLOWED/ *FORBIDDEN	Selektionsoperand – Migrations-Erlaubnis
TIMBASE	*UTC/*LTI	Selektionsoperand – Zeitbasis der Datumsangaben
TYPE	FILE	Selektionsoperand – Dateien, nicht FGG oder Dateigenerationen
	FGG	Dateigenerationen oder FGG
	PLAM	PLAM-Bibliotheken
USRINFO	*NONE/ <c-string 1..8>	Selektionsoperand – Benutzer-Metainformation
VERSION	0	Makroformat BS2000-Version < V9.5A (s. Tabelle „Versionsunterschiede VERSION=0/1/2“ auf Seite 390)
	1	Makroformat BS2000-Version V9.5A und V10.0A (s. Tabelle „Versionsunterschiede VERSION=0/1/2“ auf Seite 390)
	2	Makroformat ab Version BS2000/OSD-BC V1.0 (siehe Format und Operandenbeschreibung)
	3	Makroformat ab Version BS2000/OSD-BC V3.0 (siehe Format und Operandenbeschreibung)
VOLSET		Selektionsoperand – Volume-Set
VOLUME		Selektionsoperand – Archivnummer des Datenträgers
WORKFIL	*NO/*YES	Selektionsoperand – Arbeitsdateien

(Teil 5 von 5)

Format

Das nachfolgend dargestellte Makroaufrufformat enthält alle Operanden, die ab der Version BS2000/OSD-BC V3.0 unterstützt werden. Zur Generierung dieses Formates muss **VERSION=3** angegeben werden.

Für bestehende Programme wird Source-Kompatibilität gewährleistet, da das neue Format VERSION=3 die Funktionen der alten Formate VERSION=0/1/2 voll abdeckt. Die Änderung einer im Programm generierten Operandenliste ist jedoch nur möglich, wenn das Programm mit dem geänderten Makroaufruf neu übersetzt wird.

– Im Makroaufrufformat mit VERSION=2 können alle Operanden/Operandenwerte verwendet werden, die bis einschließlich der Version BS2000/OSD-BC V2.0A unterstützt wurden.

– Im Makroaufrufformat mit VERSION=1 können alle Operanden/Operandenwerte verwendet werden, die bis einschließlich der BS2000-Version V10.0A unterstützt wurden.

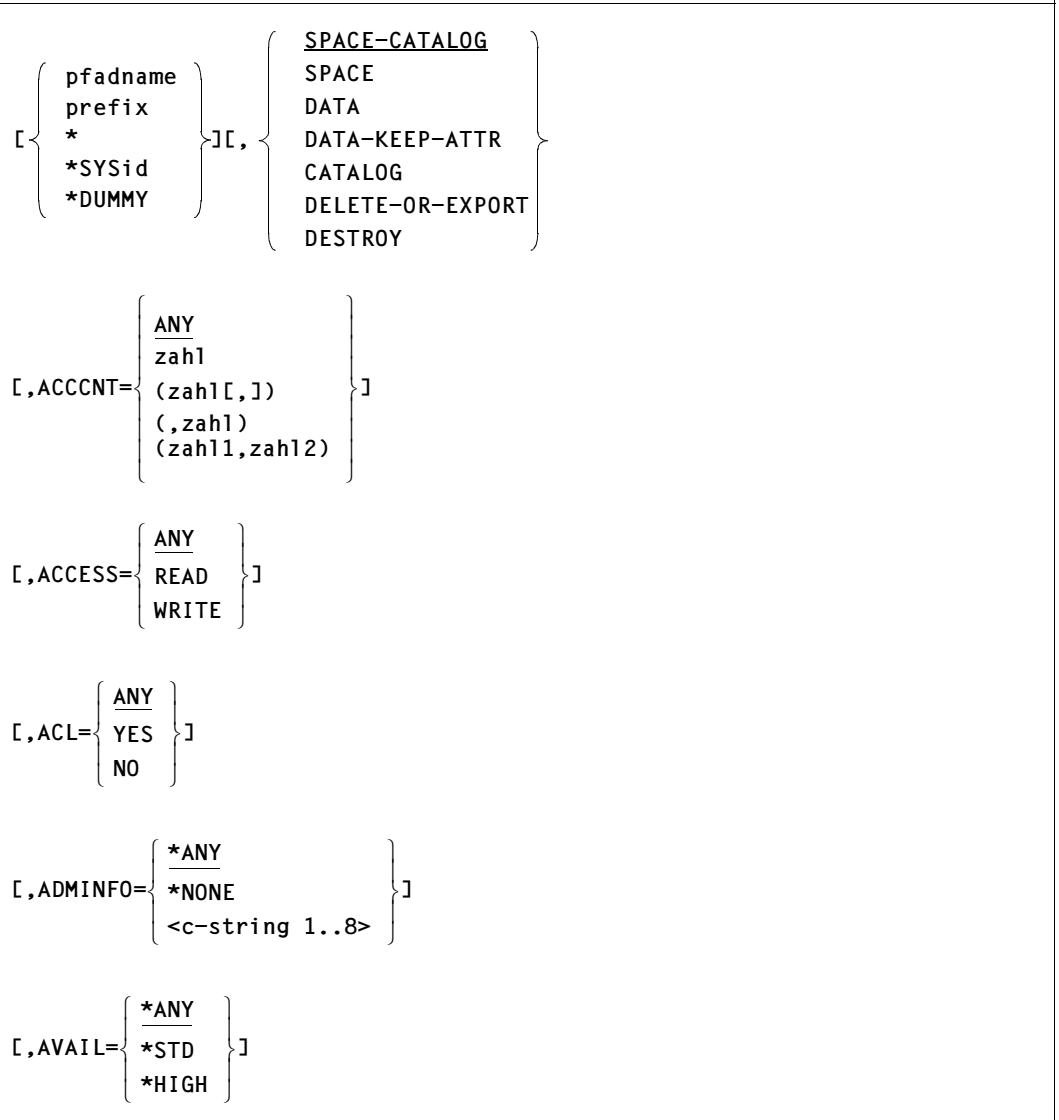
– Im Makroaufrufformat mit VERSION=0 können alle Operanden/Operandenwerte verwendet werden, die bis einschließlich der BS2000-Version V9.0A unterstützt wurden.

Die Tabelle „Versionsunterschiede VERSION=0/1/2“ auf Seite 390 zeigt, welche Operanden/Operandenwerte mit VERSION=2/1/0 unterstützt werden.

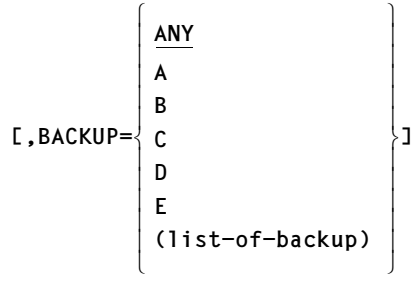
Im Format wurde die Darstellung der Operanden, denen eine Liste von Operandenwerten zugeordnet werden kann, der Übersichtlichkeit halber vereinfacht. Die Liste wird als zusätzlicher Wert in der Form `(list-of-operand)` angegeben, d.h.

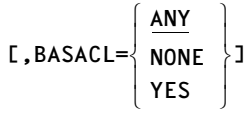
– mehrere Operandenwerte können in Listenform angegeben werden:
  $(element_1, element_2, ...)$

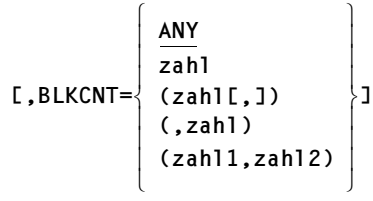
– wird nur ein Operandenwert angegeben, d.h. die Liste besteht nur aus einem Element, können die Klammern entfallen: `element` oder `(element)`.

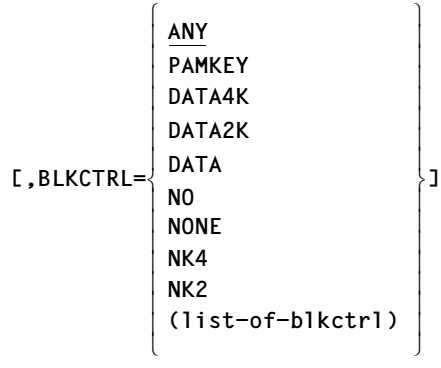
Operation	Operanden
ERASE	

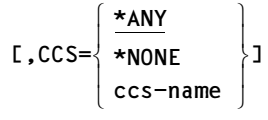
(Teil 1 von 11)

Operation	Operanden
	

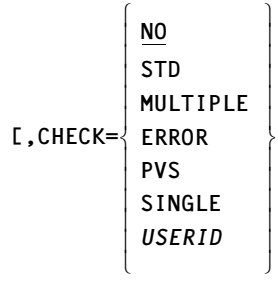
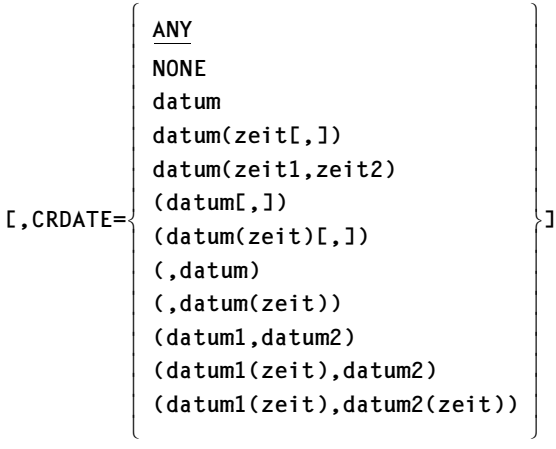
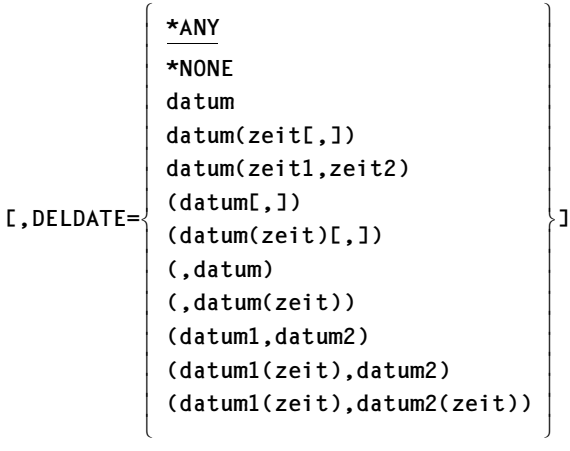




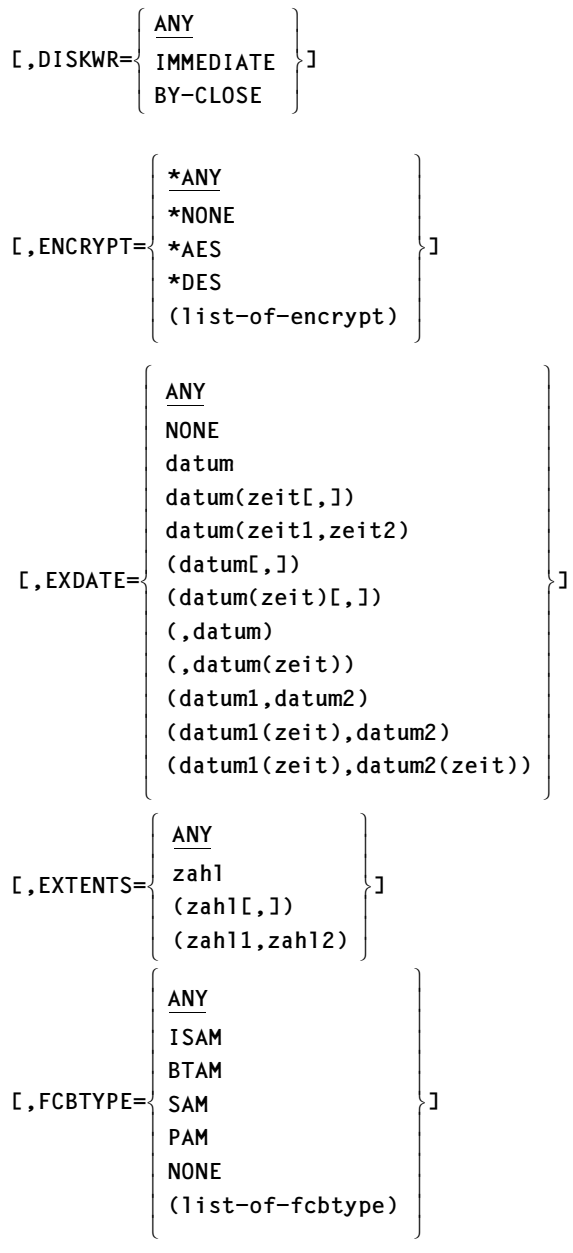




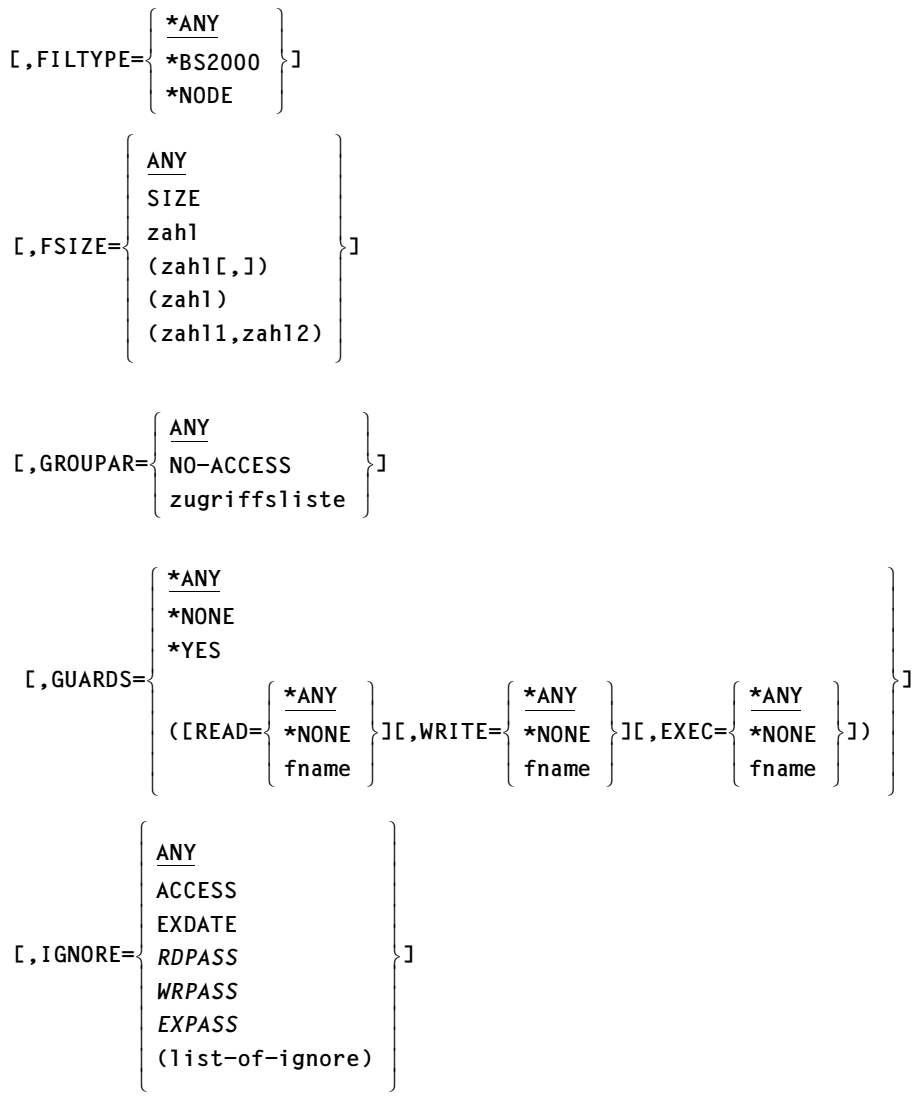
(Teil 2 von 11)

Operation	Operanden
	
	
	

(Teil 3 von 11)

Operation	Operanden
	

(Teil 4 von 11)

Operation	Operanden
	

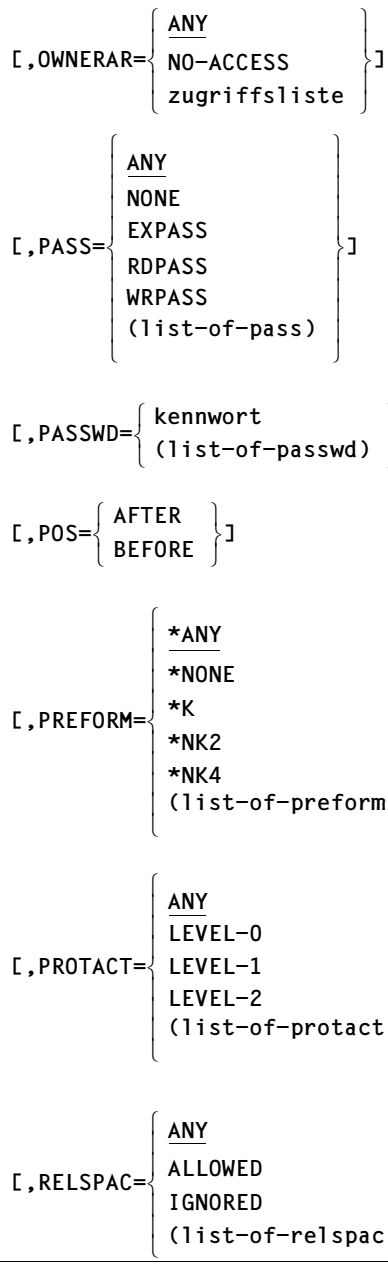
(Teil 5 von 11)

Operation	Operanden
	

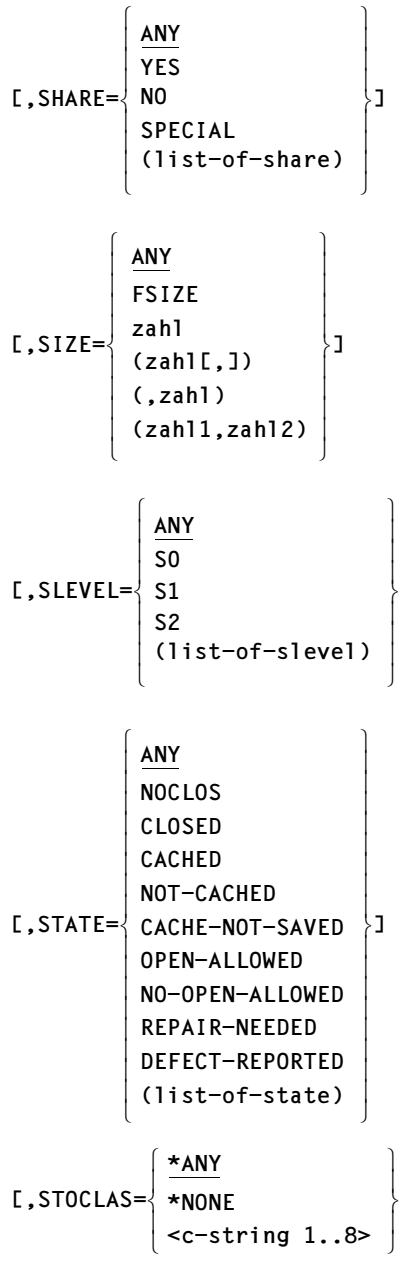
(Teil 6 von 11)

Operation	Operanden
	

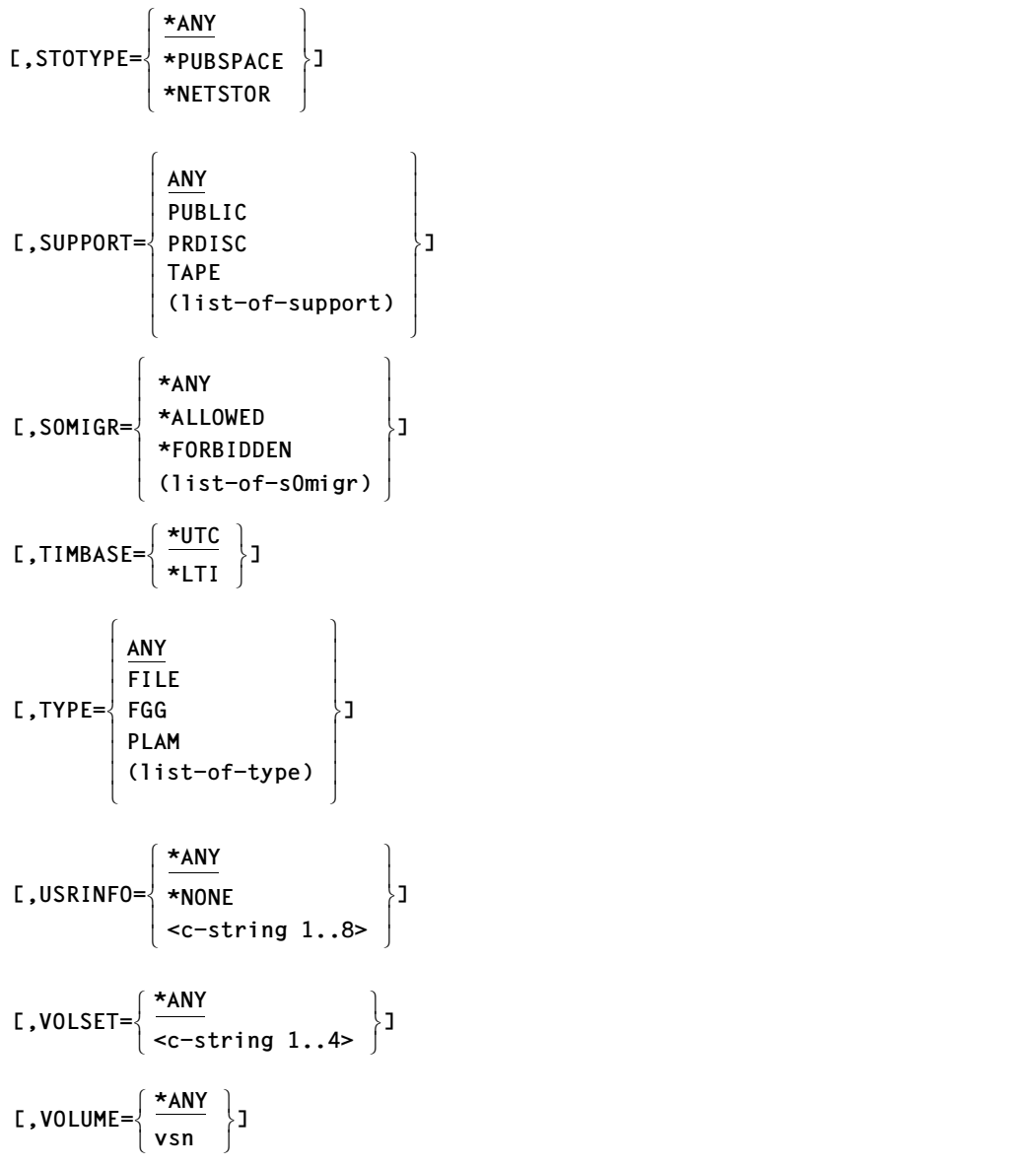
(Teil 7 von 11)

Operation	Operanden
	

(Teil 8 von 11)

Operation	Operanden
	

(Teil 9 von 11)

Operation	Operanden
	

(Teil 10 von 11)

Operation	Operanden
	[,WORKFIL={*ANY / *NO / *YES}]
	[,MF=L],VERSION={0 / 1 / 2 / 3}[,PREFIX=pre]
	MF=(E,{adr / (r)}),VERSION={0 / 1 / 2 / 3}
	MF=D,VERSION={1 / 2 / 3}[,PREFIX=pre]

(Teil 11 von 11)

Operandenbeschreibung

pfadname
bezeichnet den Pfadnamen der zu löschenden Datei(en) mit:
<c-string 1..80: filename 1..54 with-wild(80) without-gen>

Es können nur eigene Dateien oder Dateien, für die Miteigentümerschaft besteht, gelöscht werden.

Pfadname bedeutet [:catid:][$userid.][dateiname]

catid
Katalogkennung der zu löschenden Dateien; werden für „catid“ Musterzeichen verwendet, so werden diese nur für Kataloge in MPVS-Umgebung ausgewertet. Kataloge in MSCF-Umgebung können nur über die explizite „catid“ angesprochen werden (zu MSCF siehe Handbuch „HIPLEX MSCF“ [11]).
Die Default-Catid ist die der Benutzerkennung zugeordnete „catid“.

userid
Benutzerkennung; vom nichtprivilegierten Anwender können nur eigene Dateien gelöscht werden oder Dateien unter Benutzerkennungen, für die er als Mit-Eigentümer eingetragen ist. Der Systemverwalter kann auch Musterzeichen angeben.
Default-Userid: die Userid des laufenden Auftrags (d.h. des Kommandos SET-LOGON-PARAMETERS bzw. LOGON).

dateiname
bezeichnet die zu löschenden Dateien, Dateigenerationen, FGG, temporären Dateien. Der Anwender kann den Dateinamen voll- oder teilqualifiziert angeben oder Muster verwenden. Temporäre Dateien müssen mit Präfix benannt werden, sonst werden sie nicht berücksichtigt.

Wildcard-Angabe (Musterzeichen)
Der nichtprivilegierte Benutzer darf Musterzeichen nur in der Catid und im Dateinamen verwenden; der Systemverwalter auch in der Benutzerkennung (analog der Muster im FSTAT-Makro, siehe Abschnitt „Musterzeichen“ auf Seite 867).
Bei der Verwendung des Musterzeichens „*“ ist zu beachten, dass das Zeichen verdoppelt werden muss („**“), wenn mit dem Musterzeichen der Beginn des Dateinamens abgedeckt wird. Zum Beispiel löscht ERASE *SYSLST die Systemdatei SYSLST. Die Eingabe ERASE **SYSLST löscht alle Dateien, deren Dateiname mit der Zeichenfolge SYSLST endet.

prefix
Mit dem für temporäre Dateien definierten Präfix lassen sich alle temporären Dateien des Auftrags löschen. Wird der Löschvorgang protokolliert, wird der interne Name der betroffenen temporären Dateien ausgegeben.

*****

Der ERASE-Makroaufruf bezieht sich auf die EAM-Objektmoduldatei (*-Datei), die von den Übersetzern erzeugt oder verwendet wird. Außer den Kontroll- und Übersetzeroperanden werden alle übrigen Operanden auf formale Richtigkeit überprüft, sonst aber ignoriert. Fehler, die beim Löschen der *-Datei auftreten, werden ignoriert.

***SYSid**

bezeichnet die logischen Systemdateien SYSLST, SYSLSTnn und SYSOUT ($00 \leq nn \leq 99$). Für „id“ können auch Musterzeichen verwendet werden, sodass sich ein ERASE-Makroaufruf auf mehrere Systemdateien auswirken kann (Muster: siehe Abschnitt „Musterzeichen“ auf Seite 867). Mit Ausnahme der Kontroll- und Übersetzeroperanden werden die übrigen Operanden nur auf formale Richtigkeit überprüft, sonst aber ignoriert.

Die Systemdatei SYSOUT kann auch im Dialogbetrieb gelöscht werden.

Ist SYSLST einer Datei zugewiesen und wurde diese Datei mit PRINT *SYSLST ausgedruckt, werden bei einem folgenden ERASE *SYSLST nur die Seiten logisch gelöscht, die seit dem Ausdrucken erzeugt wurden.

Wenn unmittelbar nach einem ERASE *SYSOUT bzw. ERASE *SYSLST ein LOGOFF-Kommando folgt, und es ist kein Protokoll durch LIST=YES bzw. /OPTION MSG=H angefordert, so wird keine neue SYSOUT- bzw. SYSLST-Datei erstellt.

***DUMMY**

bezeichnet die Pseudodatei *DUMMY, die als „stets vorhanden“ gilt und von allen Auswahlkriterien betroffen ist. Mit Ausnahme der Kontroll- und Übersetzeroperanden werden die übrigen Operanden auf formale Richtigkeit überprüft, sonst aber ignoriert. Wird *DUMMY angegeben, sind weder Katalog- noch Datenzugriff erforderlich. *DUMMY ist vor allem von Nutzen bei Testläufen.

CATALOG

Nur für Dateien, Dateigenerationsgruppen und Dateigenerationen auf privaten Datenträgern und für Dateien auf Net-Storage:

Die Katalogeinträge der angegebenen/ausgewählten Dateien werden gelöscht, ihr Speicherplatz bleibt erhalten. Kennwortschutz wird berücksichtigt, mit ACCESS=READ oder über RETPD definierter Schreibschutz (siehe CATAL-Makro) wird jedoch ignoriert. Eine im Katalog enthaltene Vereinbarung „binär löschen“ (siehe DESTROY, CATAL-Makro) wird nicht ausgewertet.

Für Banddateien ist „CATALOG“ die Voreinstellung für die Ausführung des ERASE.

Die Aktion ERASE ...,CATALOG entspricht dem Exportieren von Dateien. Diese Dateien können später wieder importiert werden, und zwar einzeln über FILE mit der Angabe STATE=FOREIGN oder mit IMPORT, das ein oder mehrere Dateien auf Privatplatten oder auf Net-Storage gleichzeitig importieren kann. Exklusiv reservierte Dateien können nicht exportiert werden.

DATA

Nur für Plattendateien, für Banddateien gilt die Voreinstellung CATALOG:
Die betroffenen Dateien werden „logisch gelöscht“ (siehe Abschnitt „Logisch löschen“, Handbuch „Einführung in das DVS“ [1]). Anschließend sind die Daten der Datei für den Anwender nicht mehr ansprechbar, da ihm der physikalische Zugriff auf Datenträger nicht gestattet ist. Katalogeintrag und Speicherplatzzuweisung bleiben erhalten. Der Katalogeintrag ist identisch mit dem Eintrag für eine mit FILE eingerichtete Datei, die noch nicht eröffnet wurde.

DATA-KEEP-ATTR

Nur für Plattendateien, für Banddateien gilt die Voreinstellung CATALOG:
Die Dateien werden logisch gelöscht wie bei DATA, aber die datenbezogenen Eigenschaften bleiben erhalten. Die Daten selber sind für den Anwender jedoch nicht mehr ansprechbar.

DELETE-OR-EXPORT

Die ERASE-Verarbeitung wird abhängig vom Datenträgertyp, auf dem die betroffenen Dateien gespeichert sind, ausgeführt:

- Dateien, FGG usw. auf gemeinschaftlichen Datenträgern werden gelöscht, d.h. der Katalogeintrag wird gelöscht und der Speicherplatz freigegeben (entspricht der Angabe „SPACE-CATALOG“).
- Für Dateien auf Net-Storage gilt abhängig vom Dateityp:
  - Bei BS2000-Dateien wird der Katalogeintrag gelöscht und der Speicherplatz freigegeben.
  - Bei Node-Files wird der Katalogeintrag gelöscht. Die Dateien bleiben auf dem Net-Storage erhalten (entspricht dem Kommando EXPORT-NODE-FILE).
- Für Dateien, FGG usw. auf privaten Datenträgern wird nur der Katalogeintrag gelöscht (entspricht der Angabe „CATALOG“).

DESTROY
Nur für Plattendateien, bei Banddateien gilt die Voreinstellung CATALOG:
Der Speicherplatz der betroffenen Dateien wird freigegeben und der Katalogeintrag gelöscht. Zusätzlich wird der frei werdende Speicherplatz mit binär null überschrieben, sodass bei späterer Neuzuweisung des Speicherplatzes niemand die alten Daten lesen kann (Datenschutz). Für Dateien auf Privatplatte müssen dann zum Zeitpunkt des Löschens alle Datenträger der Datei bereitstehen.

„Datenzerstörung“ beim Löschen kann über CATAL im Katalogeintrag verankert werden; dort ist dann ein „DESTROY“-Kennzeichen gesetzt (DESTROY=YES). In diesem Fall wird bei Speicherplatzfreigabe der frei werdende Speicherplatz automatisch überschrieben. Wird die Datei gelöscht, werden zunächst die Aktionsoperanden ausgewertet. Soll die Datei exportiert werden (Angabe CATALOG oder DELETE-OR-EXPORT), werden die Daten nicht überschrieben, da keine Speicherplatzfreigabe erfolgt.

SPACE
Nur für Dateien auf Public-Platten und auf Net-Storage, bei Banddateien gilt die Voreinstellung CATALOG:
Der Anwender bestimmt, dass der Speicherplatz der von der ERASE-Bearbeitung betroffenen Dateien freigegeben wird. Der Katalogeintrag bleibt erhalten, wird jedoch verändert: er ist dann identisch mit einem mit CATAL erstellten Katalogeintrag. Für Dateien auf Privatplatten wird der Operand SPACE abgewiesen.

SPACE-CATALOG
ist Voreinstellung für Plattendateien. Die Katalogeinträge der betroffenen Dateien werden gelöscht und ihr Speicherplatz freigegeben.

ACCCNT
Der Anwender kann über den Zugriffszähler Dateien auswählen, die bearbeitet werden sollen. Der Zugriffszähler zeigt an, wie oft auf eine Datei zugegriffen wurde. Der Zugriffszähler kann Werte von 0 bis 2147483647 annehmen.

= ANY
Der Zugriffszähler ist kein Auswahlkriterium.

= zahl
Es werden nur Dateien bearbeitet, deren Zugriffszähler genau die angegebene Anzahl von Zugriffen anzeigt, werden bearbeitet.

= (zahl[,])
Es werden Dateien bearbeitet, deren Zugriffszähler größer oder gleich dem angegebenen Wert ist.

= (,zahl)
Es werden Dateien bearbeitet, deren Zugriffszähler kleiner oder gleich dem angegebenen Wert ist.

= (zahl1,zahl2)
Es werden Dateien bearbeitet, deren Zugriffszähler in dem angegebenen Intervall liegt (zahl1 ≤ Zugriffszähler ≤ zahl2).

ACCESS
Der Anwender kann abhängig von der Zugriffserlaubnis Dateien zur Bearbeitung auswählen.

= ANY
Die Zugriffsart ist kein Auswahlkriterium.

= READ
Es werden nur die Dateien bearbeitet, für die Schreibzugriff mit ACCESS=READ unterbunden ist, d.h. für die nur Lesezugriff zulässig ist.

= WRITE
Es werden die Dateien bearbeitet, für die Schreib- und Lesezugriff erlaubt ist.

ACL
Der Anwender kann die zu bearbeitenden Dateien danach auswählen, ob sie mit einer Zugriffsliste (ACL) geschützt sind.

= ANY
Der ACL-Eintrag ist kein Auswahlkriterium.

= NO
Es werden Dateien bearbeitet, die nicht durch einen ACL-Eintrag geschützt sind.

= YES

Seit SECOS V4.0 wird die Zugriffskontrolle über ACL nicht mehr unterstützt. Der ACL-Eintrag enthält deshalb im Normalfall den Wert NO (kein ACL-Schutz).

ADMINFO
Der Anwender kann abhängig von der Systemverwalter-Metainformation Dateien/Dateigenerationen zur Bearbeitung auswählen.

= *ANY
Die Systemverwalter-Metainformation ist kein Auswahlkriterium.

= *NONE
Es werden nur Dateien bearbeitet, die keine Systemverwalter-Metainformation besitzen.

= <c-string 1..8>
Es werden nur Dateien mit der angegebenen Systemverwalter-Metainformation bearbeitet.

AVAIL
Der Anwender kann abhängig von der Ausfallsicherheit Dateien/Dateigenerationen zur Bearbeitung auswählen.

= *ANY
Die Ausfallsicherheit ist kein Auswahlkriterium.

= *STD
Es werden nur Dateien bearbeitet, die sich nicht auf einem Volume-Set mit hoher Ausfallsicherheit befinden.

= *HIGH
Es werden nur Dateien bearbeitet, die sich auf einem Volume-Set mit hoher Ausfallsicherheit befinden (DRV-Pubset).

BACKUP
Der Anwender kann über den BACKUP-Level (Sicherungsstufe) die Dateien auswählen, die bearbeitet werden sollen. Der BACKUP-Level bestimmt, bei welchen Sicherungsläufen die Datei gesichert werden soll.

= ANY
Die Sicherungsstufe ist kein Auswahlkriterium.

= A
Es werden nur Dateien mit dem BACKUP-Level A bearbeitet

= B
Es werden nur Dateien mit dem BACKUP-Level B bearbeitet.

= C
Es werden nur Dateien mit dem BACKUP-Level C bearbeitet.

= D
Es werden nur Dateien mit dem BACKUP-Level D bearbeitet.

= E
Es werden nur Dateien mit dem BACKUP-Level E bearbeitet.

= (list-of-backup)
Es werden nur Dateien bearbeitet, die einen der angegebenen BACKUP-Level besitzen. In einer Liste können maximal 5 verschiedene BACKUP-Level angegeben werden.

BASACL
Der Anwender kann die zu bearbeitenden Dateien danach auswählen, ob sie mit einer einfachen Zugriffsliste (BASIC-ACL) geschützt sind.

= ANY
Die BASIC-ACL ist kein Auswahlkriterium.

= NONE
Es werden nur Dateien bearbeitet, für die kein BASIC-ACL-Eintrag definiert wurde.

= YES
Es werden nur Dateien bearbeitet, für die ein BASIC-ACL-Eintrag definiert wurde. Mit den Selektionsoperanden OWNERAR, GROUPAR und OTHERAR kann der Anwender die Auswahl der Dateien auf bestimmte BASIC-ACL-Einträge beschränken.

BLKCNT
Nur für Banddateien:
Der Anwender kann die zu bearbeitenden Dateien über die Anzahl der Blöcke auf Band auswählen.

= ANY
Die Anzahl der Blöcke auf Band ist kein Auswahlkriterium.

= zahl
Jede Banddatei mit genau der geforderten Anzahl von Blöcken wird bearbeitet.

= (zahl[,])
Es werden alle Banddateien bearbeitet, deren Anzahl von Blöcken größer oder gleich dem angegebenen Wert ist.

= (,zahl)
Es werden alle Banddateien bearbeitet, deren Anzahl von Blöcken kleiner oder gleich dem angegebenen Wert ist.

= (zahl1,zahl2)
Es werden alle Banddateien bearbeitet, deren Anzahl von Blöcken in dem angegebenen Intervall liegt.

Als Wert sind ganze Zahlen aus dem Intervall $0 \leq \text{wert} \leq 2147483647$ erlaubt.

BLKCTRL
Der Anwender kann über das Dateiformat die Dateien auswählen, die bearbeitet werden sollen. Das Dateiformat wird beim Erstellen der Datei festgelegt und bezieht sich auf die Existenz und Position des Blockkontrollfeldes, das die Verwaltungsinformationen enthält.

= ANY
Das Dateiformat ist kein Auswahlkriterium.

= PAMKEY
Es werden nur die Dateien bearbeitet, die für das Blockkontrollfeld einen separaten PAM-Schlüssel nutzen, d.h., die Blockkontroll-Information steht in einem separaten Schlüsselfeld außerhalb des PAM-Blockes. Die Dateien wurden mit BLKCTRL= PAMKEY erstellt (siehe FILE-Makro).

= DATA
Es werden nur die Dateien bearbeitet, bei denen das Blockkontrollfeld am Anfang des Datenblocks steht. Die Dateien wurden mit BLKCTRL=DATA erstellt (siehe FILE-Makro).

= NO
Es werden nur die Dateien bearbeitet, die kein Blockkontrollfeld enthalten. Die Dateien wurden mit BLKCTRL=NO erstellt (siehe FILE-Makro).

= NONE
Es werden nur die Dateien bearbeitet, für die kein BLKCTRL-Wert definiert wurde, d.h. Dateien, die noch nicht eröffnet wurden.

= DATA2K
Es werden nur Dateien bearbeitet, die mit BLKCTRL=DATA2K erstellt wurden (siehe FILE-Makro).

= DATA4K
Es werden nur Dateien bearbeitet, die mit BLKCTRL=DATA4K erstellt wurden (siehe FILE-Makro).

= NK2
Es werden nur NK2-Dateien bearbeitet (können sich auf NK2-Datenträgern befinden).

= NK4
Es werden nur NK4-Dateien bearbeitet (können sich auf NK4-Datenträgern befinden).

= (list-of-blkctrl)
Es werden nur Dateien bearbeitet, die einem der angegebenen Dateiformate entsprechen. Innerhalb der Liste können alle Werte außer ANY angegeben werden.

CCS
Der Anwender kann über die vereinbarte Codiertabelle (**C**oded **C**haracter **S**et) Dateien auswählen, die bearbeitet werden sollen.
In der Codiertabelle ist festgelegt, wie die Zeichen eines nationalen Zeichensatzes binär abzuspeichern sind. Der festgelegte Zeichensatz beeinflusst z.B. die Bildschirmdarstellung von Zeichen, Sortierreihenfolge usw. (siehe Handbuch „XHCS“ [22]).

= *ANY
Die Codiertabelle ist kein Kriterium für die Auswahl der Dateien, die mit ERASE bearbeitet werden sollen.

= *NONE
Nur Dateien, für die kein Zeichensatz definiert ist, werden mit ERASE bearbeitet.

= ccs-name
Es werden nur Dateien bearbeitet, für die die angegebene Codiertabelle vereinbart wurde. Der Name der Codiertabelle kann aus maximal 8 alphanumerischen Zeichen bestehen.

CHECK
Wie im Dialogbetrieb kann der Anwender vereinbaren, dass vor der Bearbeitung einer Dateimenge eine Kontrollabfrage nach SYSOUT erfolgt. Der Anwender kann bestimmen für welche Dateimenge eine Kontrollabfrage durchgeführt werden soll (z.B. für jede zu bearbeitende Datei). Eine ausgegebene Kontrollfrage muss der Anwender beantworten:

– „Y“ bestätigt, dass die angegebene Dateimenge bearbeitet werden soll.
– „N“ schließt die angegebene Dateimenge von der Bearbeitung aus.
– „T“ (Terminate) beendet die gesamte ERASE-Bearbeitung.

Eine Antwort, die nur aus Leerzeichen oder aus dem „Nullstring“ besteht, wird wie „N“ behandelt.

Im Stapelbetrieb gilt immer CHECK=NO.

= STD
Die Voreinstellung ist abhängig von der Betriebsart:

– Im interaktiven Dialog (SYSCMD ist der Datensichtstation zugeordnet) ist CHECK=MULTIPLE voreingestellt.

– In Prozeduren und im Stapelbetrieb gilt CHECK=NO.

= NO
Der Anwender kann in den Ablauf der ERASE-Verarbeitung nicht eingreifen; alle angegebenen/ausgewählten Dateien werden gelöscht.

= MULTIPLE
Ist „pfadname“ teilqualifiziert angegeben, sodass mehr als eine Datei angesprochen wird, oder enthält „pfadname“ Muster, kann der Anwender bei Wechsel der Katalogkennung entscheiden, ob Dateien aus dem jeweiligen Katalog gelöscht werden sollen. Er muss die ausgegebene „Frage“ mit „YES“ oder „NO“ beantworten. CHECK=MULTIPLE ist sinnvoll, wenn für „catid“ im „pfadname“ Muster angegeben wurden.
Im Dialog kann die ERASE-Bearbeitung abgebrochen werden (Antwort „TERMINATE“ auf die ausgegebene Frage) oder der CHECK-Modus geändert werden (nach NO/ERROR/SINGLE/PVS).

= ERROR
Der Anwender legt mit CHECK=ERROR fest, dass beim Auftreten von Fehlern, die der Aufrufer beheben kann, ein Dialog geführt wird wie bei CHECK=SINGLE; solange kein Fehler auftritt, entspricht CHECK=ERROR der Einstellung CHECK=NO (d.h. kein Dialog). CHECK=ERROR gilt implizit, wenn CHECK=SINGLE eingestellt ist.

Im Fehlerfall kann der Anwender die Fehlermeldung quittieren, die ERASE-Bearbeitung abbrechen, oder versuchen den Fehler zu beheben. Außerdem kann er den CHECK-Modus wechseln.

= PVS
Ähnlich wie bei CHECK=MULTIPLE geht die ERASE-Verarbeitung in den geführten Dialog über, wenn Dateien aus verschiedenen Katalogen betroffen sind. Der Anwender kann die „Frage“ mit „YES“ oder „NO“ beantworten, das ERASE abbrechen oder den CHECK-Modus wechseln.

= SINGLE
Der Anwender kann für jede Datei, die bearbeitet wird, im Dialog entscheiden, ob sie gelöscht werden soll oder nicht (Antwort: YES/NO). Gibt er im Dialog mit „IGNORE“ Schutzattribute an oder mit „PASSWORD“ ein oder mehrere Kennwörter, werden diese Angaben für die betreffende Datei ausgewertet und die Datei ohne weitere Rückfrage gelöscht („YES“ muss ebenfalls angegeben werden!). Der Anwender kann auch die ERASE-Verarbeitung abbrechen oder den CHECK-Modus wechseln.

Die betroffenen Dateien werden alphanumerisch sortiert aufgelistet. Sind Dateigenerationsgruppen betroffen, werden die Generationen einzeln in der Reihenfolge ihrer Generationsnummern aufgelistet. Lehnt der Anwender das Löschen einer Dateigeneration ab, wird die Bearbeitung der Dateigenerationsgruppe beendet und der aktuelle Stand gesichert (es dürfen keine Löcher in der Folge der Generationen entstehen).

Sollen nur Teile einer Generationsgruppe gelöscht werden, hängt die Reihenfolge der Generationen von der Angabe im POS-Operanden ab: für POS=AFTER werden die Generationen in absteigender Folge der Generationsnummern ausgegeben, ausgehend von der jüngsten Generation. Für POS=BEFORE ist die Reihenfolge umgekehrt – aufsteigende Generationsnummern – ausgehend von der ältesten Generation.

= *USERID*
Nur für Systemverwaltun:
Die ERASE-Verarbeitung geht in den geführten Dialog über, wenn Dateien verschiedener Benutzerkennungen betroffen sind. Bei jedem Userid-Wechsel wird abgefragt, ob die nächste Userid bearbeitet werden soll.
Die Systemverwaltung kann dies bejahen („YES“), ablehnen („NO“), die ERASE-Verarbeitung abbrechen („TERMINATE“) oder den CHECK-Modus wechseln.

CRDATE
Der Anwender kann über das Erstellungsdatum die Dateien auswählen, die bearbeitet werden sollen. Dateigenerationsgruppen und Dateigenerationen werden nicht berücksichtigt.

Datumsangaben kann der Anwender durch eine Zeitangabe ergänzen. Die Regeln für die Datums- und Zeitangaben sind auf Seite 868 beschrieben.
Bereichsangaben gelten jeweils einschließlich der angegebenen Grenzen.

= ANY
Das Erstellungsdatum ist kein Auswahlkriterium.

= NONE
Es werden nur die Dateien bearbeitet, für die noch kein Erstellungsdatum im Katalog eingetragen ist, d.h. Dateien, die noch nicht eröffnet wurden.

= datum
Es werden nur die Dateien bearbeitet, die an dem angegebenen Tag erstellt wurden.

= (datum[,])
Es werden nur Dateien bearbeitet, die seit dem angegebenen Datum erstellt wurden (Erstellungsdatum ≥ datum).

= (,datum)
Es werden nur Dateien bearbeitet, die bis zu dem angegebenen Datum erstellt wurden (Erstellungsdatum ≤ datum).

= (datum1,datum2)
Es werden nur Dateien bearbeitet, die während des angegebenen Zeitraums erstellt wurden (datum1 ≤ Erstellungsdatum ≤ datum2).

= datum(zeit[,])
Es werden nur Dateien bearbeitet, die an dem angegebenen Tag und ab der angegebenen Uhrzeit erstellt wurden.

= datum(zeit1,zeit2)
Es werden nur Dateien bearbeitet, die an dem angegebenen Tag und innerhalb des angegebenen Zeitraums erstellt wurden.

= (datum(zeit)[,])
Es werden nur Dateien bearbeitet, die ab dem angegebenen Tag und ab der angegebenen Uhrzeit erstellt wurden.

= (,datum(zeit))
Es werden nur Dateien bearbeitet, die vor dem angegebenen Tag und ab der angegebenen Uhrzeit erstellt wurden.

= (datum1(zeit),datum2(zeit))
Es werden nur Dateien bearbeitet, die in dem angegebenen Zeitraum erstellt wurden. Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.

DELDATE
Der Anwender kann über das DELETION-DATE (Zeitpunkt, ab dem die Datei ohne Berücksichtigung der Schutzattribute gelöscht werden darf) die Dateien auswählen, die bearbeitet werden sollen.

Datumsangaben kann der Anwender durch eine Zeitangabe ergänzen. Hierbei ist zu beachten, dass derzeit als Löschzeitpunkt immer die Uhrzeit 00:00:00 im Dateikatalog eingetragen ist.

Die Regeln für die Datums- und Zeitangaben sind auf Seite 868 beschrieben. Bereichsangaben gelten jeweils einschließlich der angegebenen Grenzen.

= *ANY
Das DELETION-DATE ist kein Auswahlkriterium.

= *NONE
Es werden nur die Dateien bearbeitet, für die noch kein DELETION-DATE im Katalog eingetragen ist.

= datum
Es werden nur die Dateien bearbeitet, für die das angegebene DELETION-DATE vereinbart ist.

= (datum[,])
Es werden nur Dateien bearbeitet, deren DELETION-DATE größer oder gleich dem angegebenen Datum ist.

= (,datum)
Es werden nur Dateien bearbeitet, deren DELETION-DATE kleiner oder gleich dem angegebenen Datum ist.

= (datum1,datum2)
Es werden nur Dateien bearbeitet, deren DELETION-DATE innerhalb des angegebenen Zeitraums liegt (datum1 ≤ Freigabedatum ≤ datum2).

= datum(zeit[,])
Es werden nur die Dateien bearbeitet, für die das angegebene DELETION-DATE vereinbart ist und die Uhrzeit der Freigabe größer oder gleich der angegebenen Zeit ist. Die Freigabezeit (Uhrzeit bezogen auf das DELETION-DATE) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= datum(zeit1,zeit2)
Es werden nur die Dateien bearbeitet, für die das angegebene DELETION-DATE vereinbart ist und die Uhrzeit der Freigabe innerhalb des angegebenen Zeitintervall liegt. Die Freigabezeit (Uhrzeit bezogen auf das DELETION-DATE) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (datum(zeit)[,])
Es werden nur Dateien bearbeitet, deren DELETION-DATE und Freigabezeit größer oder gleich dem angegebenen Zeitpunkt ist. Die Freigabezeit (Uhrzeit bezogen auf das DELETION-DATE) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (,datum(zeit))
Es werden nur Dateien bearbeitet, deren DELETION-DATE und Freigabezeit kleiner oder gleich dem angegebenen Zeitpunkt ist. Die Freigabezeit (Uhrzeit bezogen auf das DELETION-DATE) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (datum1(zeit),datum2(zeit))
Es werden nur Dateien bearbeitet, deren DELETION-DATE innerhalb des angegebenen Zeitraums liegt (datum1 ≤ DELETION-DATE ≤ datum2). Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.

DISKWR
Der Anwender kann die zu bearbeitenden Dateien auswählen abhängig von dem im Katalog vereinbarten Zeitpunkt, zu dem Datenkonsistenz gefordert wird.

= ANY
Der im Katalog vereinbarte Zeitpunkt, zu dem Datenkonsistenz gefordert wird, ist kein Auswahlkriterium.

= IMMEDIATE
Es werden nur Dateien bearbeitet, bei denen Datenkonsistenz direkt nach Beendigung einer Schreiboperation gefordert wird. Diese Dateien eignen sich nicht zur Bearbeitung in einem Schreibcache.

= BY-CLOSE
Es werden nur Dateien bearbeitet, bei denen Datenkonsistenz erst nach der CLOSE-Verarbeitung gefordert wird. Diese Dateien eignen sich zur Bearbeitung in einem Schreibcache.

ENCRYPT
Der Anwender kann Dateien danach auswählen, ob und mit welcher Verschlüsselungsmethode sie verschlüsselt sind.

= *ANY
Es werden alle Dateien bearbeitet, unabhängig davon, ob und mit welcher Verschlüsselungsmethode sie verschlüsselt sind.

= *NONE
Es werden nur Dateien bearbeitet, die nicht verschlüsselt sind.

= *AES
Es werden nur Dateien bearbeitet, die mit der AES-Verschlüsselungsmethode verschlüsselt sind.

= *DES
Es werden nur Dateien bearbeitet, die mit der DES-Verschlüsselungsmethode verschlüsselt sind.

EXDATE
Der Anwender kann über das Freigabedatum (Expiration Date) die Dateien auswählen, die bearbeitet werden sollen.
Das im Katalog vereinbarte Freigabedatum gibt an, ab wann die Datei erstmals wieder verändert oder gelöscht werden darf. Wird beim Erstellen der Datei kein Freigabedatum vereinbart, erhält es denselben Wert wie das Erstellungsdatum.
Dateigenerationsgruppen und Dateigenerationen werden nicht berücksichtigt.

Datumsangaben kann der Anwender durch eine Zeitangabe ergänzen. Hierbei ist zu beachten, dass derzeit als Freigabezeitpunkt immer die Uhrzeit 00:00:00 im Dateikatalog eingetragen ist.
Die Regeln für die Datums- und Zeitangaben sind auf Seite 868 beschrieben. Bereichsangaben gelten jeweils einschließlich der angegebenen Grenzen.

= ANY
Das Freigabedatum ist kein Auswahlkriterium.

= NONE
Es werden nur die Dateien bearbeitet, für die noch kein Freigabedatum im Katalog eingetragen ist, d.h. Dateien, die noch nicht eröffnet wurden.

= datum
Es werden nur die Dateien bearbeitet, für die das angegebene Freigabedatum vereinbart ist.

= (datum[,])
Es werden nur Dateien bearbeitet, deren Freigabedatum größer oder gleich dem angegebenen Datum ist.

= (,datum)
Es werden nur Dateien bearbeitet, deren Freigabedatum kleiner oder gleich dem angegebenen Datum ist.

= (datum1,datum2)
Es werden nur Dateien bearbeitet, deren Freigabedatum innerhalb des angegebenen Zeitraums liegt (datum1 ≤ Freigabedatum ≤ datum2).

= datum(zeit[,])
Es werden nur die Dateien bearbeitet, für die das angegebene Freigabedatum vereinbart ist und die Uhrzeit der Freigabe größer oder gleich der angegebenen Zeit ist. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= datum(zeit1,zeit2)
Es werden nur die Dateien bearbeitet, für die das angegebene Freigabedatum vereinbart ist und die Uhrzeit der Freigabe innerhalb des angegebenen Zeitintervall liegt. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (datum(zeit)[,])
Es werden nur Dateien bearbeitet, deren Freigabedatum und Freigabezeit größer oder gleich dem angegebenen Zeitpunkt ist. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (,datum(zeit))
Es werden nur Dateien bearbeitet, deren Freigabedatum und Freigabezeit kleiner oder gleich dem angegebenen Zeitpunkt ist. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (datum1(zeit),datum2(zeit))
Es werden nur Dateien bearbeitet, deren Freigabedatum innerhalb des angegebenen Zeitraums liegt (datum1 ≤ Freigabedatum ≤ datum2). Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.

EXTENTS
Nur für Dateien auf Platten und auf Net-Storage:
Der Anwender kann die zu bearbeitenden Dateien nach der Zahl ihrer Extents auswählen. Ein Extent ist ein zusammenhängender Bereich, den eine Datei auf einer Platte belegt. Die Anzahl der Extents, aus der eine Datei besteht ist im Katalog hinterlegt.
Eine Datei auf Net-Storage hat genau einen Extent.
Dateigenerationsgruppen und Dateigenerationen werden nicht berücksichtigt.

Für „zahl“ gilt: 0 ≤ zahl ≤ 65535; Bereichsangaben gelten jeweils einschließlich der Bereichsgrenzen.

= ANY
Die Anzahl der Extents ist kein Auswahlkriterium.

= zahl
Es werden nur Dateien mit genau der angegebenen Zahl von Extents bearbeitet.

= (zahl [,])
Es werden nur Dateien bearbeitet, die mindestens so viele Extents haben wie angegeben (Anzahl der Extents ≥ zahl).

= (,zahl)
Es werden nur Dateien bearbeitet, die höchstens so viele Extents haben wie angegeben (Anzahl der Extents ≤ zahl).

= (zahl1, zahl2)
Es werden nur Dateien bearbeitet, die mindestens so viele Extents haben wie „zahl1“ und höchstens so viele wie „zahl2“ (zahl1 ≤ Anzahl der Extents ≤ zahl2).

FCBTYPE
Der Anwender kann die zu bearbeitenden Dateien auswählen über die Zugriffsmethode, mit der die Dateien erstellt wurden. Die Zugriffmethode bei Dateierstellung ist im Katalog eingetragen. Sie entspricht der Angabe für FCBTYPE im FILE-Makro.
Dateigenerationsgruppen und Dateigenerationen werden nicht berücksichtigt.

= ANY
Die Zugriffsmethode ist kein Auswahlkriterium.

= NONE
Es werden nur Dateien bearbeitet, für die noch keine Zugriffsmethode im Katalog eingetragen ist, d.h. Dateien, die noch nicht eröffnet wurden.

= ISAM
Es werden Dateien bearbeitet, die mit der Zugriffsmethode ISAM erstellt wurden (ISAM-Dateien).

= BTAM
Es werden Dateien bearbeitet, die mit der Zugriffsmethode BTAM erstellt wurden (BTAM-Dateien). BTAM-Dateien sind Banddateien. Sie können deshalb nur exportiert werden (siehe Aktionsoperand CATALOG).

= SAM
Es werden Dateien bearbeitet, die mit der Zugriffsmethode SAM erstellt wurden (SAM-Dateien).

= PAM
Es werden Dateien bearbeitet, die mit der Zugriffsmethode UPAM erstellt wurden (PAM-Dateien).

= (list-of-fcbtype)
In einer Liste können mehrere Zugriffsmethoden angegeben werden. Es werden Dateien bearbeitet, die mit einer der angegebenen Zugriffsmethoden erstellt wurden.

FILTYPE
Der Anwender kann die zu bearbeitenden Dateien nach dem Dateityp auswählen.

= *ANY
Der Dateityp ist kein Auswahlkriterium.

= *BS2000
Es werden BS2000-Dateien bearbeitet.

= *NODE
Es werden Dateien bearbeitet, die als Node-Files angelegt sind.

FSIZE
Nur für Dateien auf Platte und auf Net-Storage:
Der Anwender kann die zu bearbeitenden Dateien auswählen abhängig von Anzahl der freien PAM-Seiten. Die freien PAM-Seiten einer Datei bezeichnen die Größe des reservierten, aber nicht belegten Speicherplatzes.
Dateigenerationsgruppen und Dateigenerationen werden nicht berücksichtigt.

Für „zahl“ gilt: $0 \leq zahl \leq 2147483647$;
Bereichsangaben gelten jeweils einschließlich der Bereichsgrenzen.

= ANY
Die Größe des freien (= reservierten, aber nicht belegten) Speicherplatzes ist kein Auswahlkriterium.

= SIZE
Es werden nur Dateien bearbeitet, bei denen keine der reservierten Seiten belegt ist (d.h. es ist keine PAM-Seite beschrieben).

= zahl
Es werden nur Dateien bearbeitet, die genau so viele freie PAM-Seiten haben wie angegeben.

= (zahl[,])
Es werden nur Dateien bearbeitet, die mindestens so viele freie PAM-Seiten haben wie angegeben (freie PAM-Seiten $\geq$ zahl).

= (,zahl)
Es werden nur Dateien bearbeitet, die höchstens so viele freie PAM-Seiten haben wie angegeben (freie PAM-Seiten $\leq$ zahl).

= (zahl1, zahl2)
Es werden nur Dateien bearbeitet, die mindestens so viele freie PAM-Seiten haben wie „zahl1“ und höchstens so viele wie „zahl2“ (zahl1 $\leq$ freie PAM-Seiten $\leq$ zahl2).

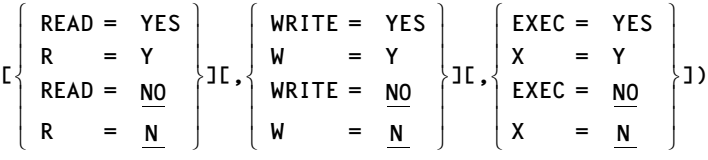
GROUPAR
Der Anwender kann die zu bearbeitenden Dateien in Abhängigkeit von den Zugriffsrechten auswählen, die in ihren BASIC-ACL-Einträgen für die Mitglieder der Benutzergruppe des Dateieigentümers festgelegt sind.

= ANY
Die BASIC-ACL-Einträge für die Mitglieder der Benutzergruppe des Dateieigentümers sind kein Auswahlkriterium.

= NO-ACCESS
Es werden nur Dateien bearbeitet, auf die die Benutzergruppe des Eigentümers nicht zugreifen darf.

= zugriffsliste
Es werden nur Dateien bearbeitet, in deren BASIC-ACL-Eintrag für die Benutzergruppe des Dateieigentümers mindestens eines der in der Liste angegebenen Zugriffsrechte vereinbart ist.

zugriffsliste hat folgendes Format:



Die runden Klammern sind Bestandteil des Operandenwertes und müssen mit angegeben werden. Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=YES bzw. R=Y	Bearbeitet alle Dateien, auf die die Benutzergruppe des Eigentümers lesend zugreifen darf.
READ=NO bzw. R=N	Bearbeitet alle Dateien, auf die die Benutzergruppe des Eigentümers nicht lesend zugreifen darf.
WRITE=YES bzw. W=Y	Bearbeitet alle Dateien, auf die die Benutzergruppe des Eigentümers schreibend zugreifen darf.
WRITE=NO bzw. W=N	Bearbeitet alle Dateien, auf die die Benutzergruppe des Eigentümers nicht schreibend zugreifen darf.
EXEC=YES bzw. X=Y	Bearbeitet alle Dateien, die die Benutzergruppe des Eigentümers ausführen darf.
EXEC=NO bzw. X=N	Bearbeitet alle Dateien, die die Benutzergruppe des Eigentümers nicht ausführen darf.

GUARDS
Der Anwender kann die zu bearbeitenden Dateien auswählen abhängig von einem vereinbarten Zugriffsschutz mit GUARDS (siehe Handbuch „SECOS“ [8]).

= *ANY
Der vereinbarte Zugriffsschutz mit GUARDS ist kein Auswahlkriterium.

= *NONE
Es werden alle Dateien bearbeitet, die keinen Zugriffsschutz über GUARDS definiert haben.

= *YES
Es werden alle Dateien bearbeitet, die einen Zugriffsschutz über GUARDS definiert haben.

= (READ=...,WRITE=...,EXEC=...)
Innerhalb einer Liste kann der Anwender angeben, wie der Zugriffschutz mit GUARDS für die auszuwählenden Dateien vereinbart sein soll. Für jede Zugriffsart (Lesen, Schreiben und Ausführen) kann der vereinbarte Schutz genau angegeben werden. Wird für eine Zugriffsart keine Angabe gemacht, so erfolgt die Auswahl unabhängig von dem dafür vereinbarten Schutz.

Je Zugriffsart kann einer der folgenden Werte angegeben werden:

*ANY	Der vereinbarte GUARDS-Schutz ist kein Auswahlkriterium.
*NONE	Für die angegebene Zugriffsart ist kein Guard vereinbart, d.h. der entsprechende Zugriff wird untersagt.
fname	Für die angegebene Zugriffsart sind im Guard fname alle Bedingungen für die Zugriffserlaubnis enthalten.

IGNORE
Der Anwender kann bestimmen, ob ein vereinbarter Schutz gegen Schreibzugriffe oder eine vereinbarte Schutzfrist ignoriert werden sollen.
Die Angabe des Operanden IGNORE ersetzt den Aufruf des CATAL-Makros, mit dem gegebenenfalls die Schutzattribute für die zu bearbeitenden Dateien vor Aufruf des ERASE-Makros zurückgesetzt werden müssten.

= ANY
Ist kein Operandenwert angegeben, sollen alle angegebenen Schutzattribute bei der ERASE-Bearbeitung beachtet werden.

= ACCESS
Dateien, die gegen Schreibzugriff des Eigentümers geschützt sind, können bei der ERASE-Verarbeitung gelöscht werden. Ein bestehender Schreibschutz wird ignoriert. Die Angabe wird ignoriert, wenn für eine Datei unter fremder Benutzerkennung eine TSOS-Einschränkung vorliegt.

= EXDATE
Dateien, für die noch eine Schutzfrist besteht (Freigabedatum > aktueller Zeitpunkt), können bei der ERASE-Verarbeitung gelöscht werden. Eine bestehende Schutzfrist wird ignoriert.

= *RDPASS*
Nur für Systemverwaltung:
Dateien, die mit einem Lesekennwort geschützt sind, können bei der ERASE-Verarbeitung gelöscht werden. Ein bestehender Zugriffsschutz über ein Lesekennwort wird ignoriert.

= *WRPASS*
Nur für Systemverwaltung:
Dateien, die mit einem Schreibkennnwort geschützt sind, können bei der ERASE-Verarbeitung gelöscht werden. Ein bestehender Zugriffsschutz über ein Schreibkennwort wird ignoriert.

= *EXPASS*
Nur für Systemverwaltung:
Dateien, die mit einem Ausführungskennwort geschützt sind, können bei der ERASE-Verarbeitung gelöscht werden. Ein bestehender Zugriffsschutz über ein Ausführungskennwort wird ignoriert.

= (list-of-ignore)
In einer Liste können die Operandenwerte ACCESS und IGNORE angegeben werden, d.h. beide Schutzattribute werden ignoriert.

IOPERF
Der Anwender kann die zu bearbeitenden Dateien auswählen abhängig von dem Performance-Attribut, das im Katalog vereinbart wurde (siehe Operand IOPERF, CATAL-Makro).

= ANY
Die Performance-Eigenschaft ist kein Auswahlkriterium.

= STD
Es werden nur Dateien bearbeitet, deren Performance-Attribut mit STD vereinbart wurde.

= HIGH
Es werden nur Dateien bearbeitet, deren Performance-Attribut mit HIGH vereinbart wurde (hohe Performance-Priorität).

= VERY-HIGH
Es werden nur Dateien bearbeitet, deren Performance-Attribut mit VERY-HIGH vereinbart wurde (höchste Performance-Priorität).

= (list-of-ioperf)
In einer Liste können bis zu drei Performance-Attribute angegeben werden. Es werden nur Dateien bearbeitet, die eines der angegebenen Attribute besitzen.

IOUSAGE
Der Anwender kann die zu bearbeitenden Dateien auswählen abhängig von der Art der Ein/Ausgabe-Operationen, auf die sich Performance-Attribut bezieht (siehe Operand IOUSAGE, CATAL-Makro).

= ANY
Das Performance-Attribut ist kein Auswahlkriterium.

= RDWRT
Es werden nur Dateien bearbeitet, deren Performance-Attribut sich auf Schreib- und Leseoperationen bezieht.

= WRITE
Es werden nur Dateien bearbeitet, deren Performance-Attribut sich auf Schreiboperationen bezieht.

= READ
Es werden nur Dateien bearbeitet, deren Performance-Attribut sich auf Leseoperationen bezieht.

= (list-of-iousage)
In einer Liste können mehrere Arten von Ein/Ausgabe-Operationen angegeben werden. Es werden nur Dateien bearbeitet, deren Performance-Attribut sich auf eine der angegebenen Ein/Ausgabe-Operationen bezieht.

KEEPACL
Nur für Systemverwaltung:
Der Anwender bestimmt, ob beim Löschen des Katalogeintrags einer ACL-geschützten Datei auch die Zugriffsliste (ACL) gelöscht werden soll.

= *NO
ist Voreinstellung. Eine ACL-Zugriffsliste wird zusammen mit dem Katalogeintrag gelöscht.

= YES
Eine Zugriffsliste bleibt beim Löschen des Katalogeintrags erhalten.

Seit SECOS V4.0 wird die Zugriffskontrolle über ACL nicht mehr unterstützt.

LADATE
Der Anwender kann über das Datum des letzten Zugriffs die Dateien auswählen, die bearbeitet werden sollen. Dateigenerationsgruppen und Dateigenerationen werden nicht berücksichtigt.

Datumsangaben kann der Anwender durch eine Zeitangabe ergänzen. Die Regeln für die Datums- und Zeitangaben sind auf Seite 868 beschrieben.
Bereichsangaben gelten jeweils einschließlich der angegebenen Grenzen.

= ANY
Das Datum des letzten Zugriffs ist kein Auswahlkriterium.

= NONE
Es werden nur die Dateien bearbeitet, für die noch kein Zugriffsdatum im Katalog eingetragen ist, d.h. Dateien, die noch nicht eröffnet wurden.

= datum
Es werden nur die Dateien bearbeitet, auf die an dem angegebenen Tag zuletzt zugegriffen wurde.

= (datum[,])
Es werden nur Dateien bearbeitet, auf die ab dem angegebenen Datum zuletzt zugegriffen wurde (letzter Zugriff ≥ datum).

= (,datum)
Es werden nur Dateien bearbeitet, auf die bis zu dem angegebenen Datum zuletzt zugegriffen wurde (letzter Zugriff ≤ datum).

= (datum1,datum2)
Es werden nur Dateien bearbeitet, auf die während des angegebenen Zeitraums zuletzt zugegriffen wurde (datum1 ≤ letzter Zugriff ≤ datum2).

= datum(zeit[,])
Es werden nur Dateien bearbeitet, auf die an dem angegebenen Tag und ab der angegebenen Uhrzeit zuletzt zugegriffen wurde.

= datum(zeit1,zeit2)
Es werden nur Dateien bearbeitet, auf die an dem angegebenen Tag und innerhalb des angegebenen Zeitraums zuletzt zugegriffen wurde.

= (datum(zeit)[,])
Es werden nur Dateien bearbeitet, auf die ab dem angegebenen Tag und ab der angegebenen Uhrzeit zuletzt zugegriffen wurde.

= (,datum(zeit))
Es werden nur Dateien bearbeitet, auf die vor dem angegebenen Tag und der angegebenen Uhrzeit zuletzt zugegriffen wurde.

= (datum1(zeit),datum2(zeit))
Es werden nur Dateien bearbeitet, auf die in dem angegebenen Zeitraum zuletzt zugegriffen wurde. Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.

LASTPAG
Der Anwender kann über die Anzahl der belegten PAM-Seiten die Dateien auswählen, die bearbeitet werden sollen, Der Last Page Pointer zeigt die höchste belegte PAM-Seite.

= ANY
Der belegte Speicherplatz ist kein Auswahlkriterium.

= zahl
Es werden nur Dateien bearbeitet, die genau die angegebene Anzahl von PAM-Seiten belegen.

= (zahl[,])
Es werden nur Dateien bearbeitet, deren Anzahl belegter Seiten größer oder gleich dem angegebenen Wert ist.

= (,zahl)
Es werden nur Dateien bearbeitet, deren Anzahl belegter Seiten kleiner oder gleich dem angegebenen Wert ist.

= (zahl1,zahl2)
Es werden nur Dateien bearbeitet, deren Anzahl belegter Seiten in dem angegebenen Intervall liegt (zahl1 ≤ belegte PAM-Seiten ≤ zahl2).

Als Wert sind ganze Zahlen aus dem Intervall 0 ≤ zahl ≤ 2147483647 erlaubt.

LCDATE
Der Anwender kann über das Datum des letzten Schreibzugriffs die Dateien auswählen, die bearbeitet werden sollen. Dateigenerationsgruppen und Dateigenerationen werden nicht berücksichtigt.

Datumsangaben kann der Anwender durch eine Zeitangabe ergänzen. Die Regeln für die Datums- und Zeitangaben sind auf Seite 868 beschrieben.
Bereichsangaben gelten jeweils einschließlich der angegebenen Grenzen.

= ANY
Das Datum des letzten Schreibzugriffs ist kein Auswahlkriterium.

= NONE
Es werden nur die Dateien bearbeitet, für die noch kein Datum des letzten Schreibzugriffs im Katalog eingetragen ist, d.h. Dateien, die noch nicht eröffnet wurden.

= datum
Es werden nur die Dateien bearbeitet, auf die an dem angegebenen Tag zuletzt schreibend zugegriffen wurde.

= (datum[,])
Es werden nur Dateien bearbeitet, auf die ab dem angegebenen Datum zuletzt schreibend zugegriffen wurde (letzter Zugriff ≥ datum).

= (,datum)
Es werden nur Dateien bearbeitet, auf die bis zu dem angegebenen Datum zuletzt schreibend zugegriffen wurde (letzter Zugriff ≤ datum).

= (datum1,datum2)
Es werden nur Dateien bearbeitet, auf die während des angegebenen Zeitraums zuletzt schreibend zugegriffen wurde (datum1 ≤ letzter Zugriff ≤ datum2).

= datum(zeit[,])
Es werden nur Dateien bearbeitet, auf die an dem angegebenen Tag und ab der angegebenen Uhrzeit zuletzt schreibend zugegriffen wurde.

= datum(zeit1,zeit2)
Es werden nur Dateien bearbeitet, auf die an dem angegebenen Tag und innerhalb des angegebenen Zeitraums zuletzt schreibend zugegriffen wurde.

= (datum(zeit)[,])
Es werden nur Dateien bearbeitet, auf die ab dem angegebenen Tag und ab der angegebenen Uhrzeit zuletzt schreibend zugegriffen wurde.

= (,datum(zeit))
Es werden nur Dateien bearbeitet, auf die bis zu dem angegebenen Tag und der angegebenen Uhrzeit zuletzt schreibend zugegriffen wurde.

= (datum1(zeit),datum2(zeit))
Es werden nur Dateien bearbeitet, auf die in dem angegebenen Zeitraum zuletzt schreibend zugegriffen wurde. Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.

LIST
Der Anwender kann bestimmen, ob der Ablauf des ERASE-Makroaufrufs protokolliert wird oder ob Fehler, die bei der Bearbeitung einer Datei auftreten, nach SYSOUT gemeldet werden. Fehler werden sonst nur über das Returncode-Feld zurückgemeldet.

Voreinstellung: es wird kein zusätzliches Protokoll erzeugt

= ERRORS-TO-SYSOUT
Fehler werden nach SYSOUT protokolliert (Ausnahme: über NOSTEP ignorierte Fehler).

= (area,len)
Die Namen aller vom ERASE betroffenen Dateien werden in einen Benutzer-Ausgabebereich geschrieben.

```
(area,len) bedeutet: ( { adr  } , { länge } )
                       { (r1) }   { equ    }
                                  { (r2)  }
```

adr	symbolische Adresse des Ausgabebereichs
r1	Register r1 enthält die Adresse des Ausgabebereichs
länge	Länge des Ausgabebereichs als Konstante
equ	Länge des Ausgabebereichs als Equate
r2	Register r2 enthält die Länge des Ausgabereichs

In den Ausgabebereich werden nacheinander Einzelinformationen geschrieben, die folgenden Aufbau haben:

SL	D	RC	pfadname	EK

SL	Satzlänge (Feldlänge: 2 Byte), zeigt Länge einer Einzelinformation an
D	2 Byte, reserviert
RC	Rückinformation (Feldlänge: 4 Byte): DMS-Meldungsschlüssel, der mit Hilfe des IDEMS-Makroaufrufs ausgewertet werden kann
pfadname	Pfadname der Datei (Feldlänge: variabel)
EK	Endekriterium (Feldlänge: 2 Byte)

Das Ende der Ausgabe im Ausgabebereich wird anzeigt mit X'00' im Feld EK. Bei Überlauf des Ausgabebereichs wird Subcode2 im Fehlercode auf „1“ gesetzt, die Verarbeitung wird ohne Protokollierung fortgesetzt.

Da die internen Pufferbereiche beschränkt sind, kann bei RFA-Zugriff ein Überlauf angezeigt werden, obwohl der Anwender einen genügend großen Ausgabebereich zur Verfügung gestellt hat. Die Ausgabe wurde dann vorzeitig abgebrochen.

MANCLAS
Der Anwender kann die zu bearbeitenden Dateien entsprechend der HSMS-Management-Klasse zur Dateisicherung auf SM-Pubsets auswählen.

= *ANY
Die HSMS-Management-Klasse ist kein Auswahlkriterium.

= *NONE
Nur Dateien, für die keine HSMS-Management-Klasse definiert ist, werden ausgewählt.

= <c-string 1..8>
Nur Dateien mit der angegebenen HSMS-Management-Klasse werden ausgewählt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.
In allen Makroaufrufen, die sich durch den MF-Operanden unterscheiden (MF=L/E/D), muss der Versionsoperand den gleichen Wert haben.
Die vorliegende Beschreibung enthält alle Operanden, die mit BS2000/OSD-BC ≥ V3.0 unterstützt werden. Zur Generierung dieses Formates muss VERSION=3 angegeben werden.

MIGRATE
Der Anwender kann über die im Katalog vereinbarte Migration (siehe Makro CATAL, Operand MIGRATE, Seite 170) die Dateien auswählen, die bearbeitet werden sollen.

= ANY
Die angegebenen Dateien werden unabhängig von der jeweiligen MIGRATE-Vereinbarung im Katalogeintrag bearbeitet.

= ALLOWED
Es werden nur Dateien bearbeitet, für die im Katalogeintrag der entsprechende Operandenwert vereinbart wurde, d.h. Dateien, die auf die Speicherebenen S1 und S2 verdrängt werden dürfen.

= INHIBIT
Es werden nur Dateien bearbeitet, für die im Katalogeintrag MIGRATE=INHIBIT vereinbart wurde, d.h. Dateien, die kurzfristig (z.B. für Reorganisationszwecke) verdrängt werden dürfen.

= FORBIDDEN
Es werden nur Dateien bearbeitet, für die im Katalogeintrag der entsprechende Operandenwert vereinbart wurde, d.h. Dateien, die nicht verdrängt werden dürfen.

= (list-of-migrate)
Der Anwender kann die Werte ALLOWED und INHIBIT in einer Liste angeben. Es werden nur Dateien bearbeitet, für die im Katalog einer der angegebenen Werte vereinbart wurde.

MOUNT

Nur für Dateien auf Privatplatten:

Der Anwender gibt an, ob zu Beginn der ERASE-Bearbeitung einer Datei auf Privatplatte nur die erste oder alle betroffenen Privatplatten online sein müssen.
Der MOUNT-Operand sollte zusammen mit den Operanden SPACE-CATALOG oder DESTROY angegeben werden; in Kombination mit DATA muss er angegeben werden.

Für Banddateien oder Dateien auf Public-Platten wird eine MOUNT-Angabe ignoriert.

Voreinstellung: MOUNT = FIRST-DISK

= FIRST-DISK
Nur die Privatplatte, auf der die Datei beginnt und die den Katalogeintrag der Datei enthält, muss online sein.

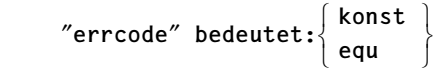
= ALL-DISK
Alle Privatplatten, auf denen Teile der Datei gespeichert sind, müssen online sein. Fehlt eine Platte, wird die Datei nicht gelöscht.

NOSTEP

Der Anwender gibt an, welche Fehler nicht als Rückinformation in Register 15 gemeldet werden sollen.

= NONE
Alle Fehler sollen zurückgemeldet werden.

= errcode
Der Anwender kann über den DVS-Fehlerschlüssel definieren, welche Fehler ignoriert werden, d.h. nicht über das Returncode-Feld im Standardheader zurückgemeldet werden sollen.



konst der Fehlerschlüssel wird als Konstante dezimal oder hexadezimal angegeben
equ der Fehlerschlüssel wird als Equate angegeben

Es wird empfohlen, die Equates zu verwenden, die der IDEMS-Makroaufruf erzeugt.

Die Liste (errcode,...) kann maximal drei Elemente enthalten.

OTHERAR
Der Anwender kann die zu bearbeitenden Dateien in Abhängigkeit von den Zugriffsrechten auswählen, die in ihren BASIC-ACL-Einträgen für alle Anwender außerhalb der Benutzergruppe des Dateieigentümers festgelegt sind.

= ANY
Die BASIC-ACL-Einträge für alle Anwender außerhalb der Benutzergruppe des Dateieigentümers sind kein Auswahlkriterium.

= NO-ACCESS
Es werden nur Dateien bearbeitet, auf die Anwender außerhalb der Benutzergruppe des Eigentümers nicht zugreifen dürfen.

= zugriffsliste
Es werden nur Dateien bearbeitet, in deren BASIC-ACL-Eintrag für Anwender außerhalb der Benutzergruppe des Dateieigentümers mindestens eines der in der Liste angegebenen Zugriffsrechte vereinbart ist.

„zugriffsliste“ hat folgendes Format:

```
([ READ = YES ]  [ WRITE = YES ]  [ EXEC = YES ])
 [ R    = Y   ]  [ W     = Y   ]  [ X    = Y   ]
 [ READ = NO  ]  [ WRITE = NO  ]  [ EXEC = NO  ]
 [ R    = N   ]  [ W     = N   ]  [ X    = N   ]
```

Die runden Klammern sind Bestandteil des Operandenwertes und müssen mit angegeben werden. Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=YES bzw. R=Y	Bearbeitet alle Dateien, auf die Anwender außerhalb der Benutzergruppe des Eigentümers lesend zugreifen dürfen.
READ=NO bzw. R=N	Bearbeitet alle Dateien, auf die Anwender außerhalb der Benutzergruppe des Eigentümers nicht lesend zugreifen dürfen.
WRITE=YES bzw. W=Y	Bearbeitet alle Dateien, auf die Anwender außerhalb der Benutzergruppe des Eigentümers schreibend zugreifen dürfen.
WRITE=NO bzw. W=N	Bearbeitet alle Dateien, auf die Anwender außerhalb der Benutzergruppe des Eigentümers nicht schreibend zugreifen dürfen.
EXEC=YES bzw. X=Y	Bearbeitet alle Dateien, die Anwender außerhalb der Benutzergruppe des Eigentümers ausführen dürfen.
EXEC=NO bzw. X=N	Bearbeitet alle Dateien, die Anwender außerhalb der Benutzergruppe des Eigentümers nicht ausführen dürfen.

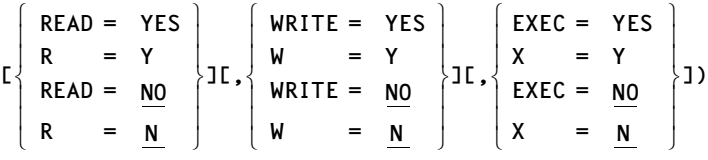
OWNERAR
Der Anwender kann die zu bearbeitenden Dateien in Abhängigkeit von den Zugriffsrechten auswählen, die in ihren BASIC-ACL-Einträgen für die Dateieigentümer festgelegt sind.

= ANY
Die BASIC-ACL-Einträge für die Dateieigentümer sind kein Auswahlkriterium.

= NO-ACCESS
Es werden nur Dateien bearbeitet, auf die der Eigentümer nicht zugreifen darf.

= zugriffsliste
Es werden nur Dateien bearbeitet, in deren BASIC-ACL-Eintrag für den Dateieigentümer mindestens eines der in der Liste angegebenen Zugriffsrechte vereinbart ist.

„zugriffsliste“ hat folgendes Format:



Die runden Klammern sind Bestandteil des Operandenwertes und müssen mit angegeben werden. Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=YES bzw. R=Y	Bearbeitet alle Dateien, auf die der Eigentümer lesend zugreifen darf.
READ=NO bzw. R=N	Bearbeitet alle Dateien, auf die der Eigentümer nicht lesend zugreifen darf.
WRITE=YES bzw. W=Y	Bearbeitet alle Dateien, auf die der Eigentümer schreibend zugreifen darf.
WRITE=NO bzw. W=N	Bearbeitet alle Dateien, auf die der Eigentümer nicht schreibend zugreifen darf.
EXEC=YES bzw. X=Y	Bearbeitet alle Dateien, die der Eigentümer ausführen darf.
EXEC=NO bzw. X=N	Bearbeitet alle Dateien, die der Eigentümer nicht ausführen darf.

PASS
Der Anwender kann über den Kennworttyp die Dateien auswählen, die vom ERASE bearbeitet werden sollen.

= ANY
Der Kennwortschutz ist kein Auswahlkriterium.

= NONE
Es werden nur Dateien bearbeitet, für die kein Kennwortschutz definiert wurde.

= EXPASS
Es werden nur Dateien bearbeitet, die durch ein Ausführungskennwort geschützt sind.

= RDPASS
Es werden nur Dateien bearbeitet, die durch ein Lesekennwort geschützt sind.

= WRPASS
Es werden nur Dateien bearbeitet, die durch ein Schreibkennwort geschützt sind.

= (list-of-pass)
In einer Liste kann der Anwender mehrere Arten des Kennwortschutzes angeben. Es werden nur Dateien bearbeitet, die mit einem der angegebenen Kennworttypen geschützt sind.

PASSWD
Der Anwender kann ein oder mehrere Kennwörter angeben, sodass die durch diese Kennwörter geschützten Dateien gelöscht werden können. Die angegebenen Kennwörter müssen nicht in die Kennworttabelle des Auftrags eingetragen sein. Die angegebenen Kennwörter gelten jedoch nur für die aktuelle ERASE-Verarbeitung.

Die Kennwortangabe muss den Regeln für die Kennwort-Definition entsprechen, in Protokollen treten Kennwörter nicht im Klartext auf.

Wird keine Angabe gemacht, so werden von ERASE keine mit Kennwort geschützten Dateien bearbeitet.

= ANY
Dem ERASE wird kein Kennwort mitgeteilt.

= kennwort
Der Schutz durch dieses Kennwort soll entfallen.

= (list-of-passwd)
Der Anwender kann maximal 3 Kennwörter in einer Liste angeben.

POS

Nur für Dateigenerationen:

In „pfadname“ können außer in der Generationsnummer Muster verwendet werden. Die Generationsnummer muss als absolute oder relative Generationsnummer angegeben werden. Die mit „pfadname“ identifizierte Generation muss existieren und wird nicht gelöscht.

Abhängig vom Operandenwert AFTER/BEFORE werden alle jüngeren oder alle älteren Dateigenerationen gelöscht. Der Katalogeintrag wird aktualisiert:

– Wird die älteste Generation gelöscht, so wird die in „pfadname“ angegebene Generation zur ältesten Generation.
– Wird die jüngste Generation gelöscht, so wird die in „pfadname“ angegebene Generation zur jüngsten Generation.
– Wird die Generation mit der relativen Generationsnummer 0 gelöscht, so wird die in „pfadname“ angegebene Generation wird zur Basisgeneration.

= AFTER

Alle durch „pfadname“ ausgewählten Generationen, deren Generationsnummer größer ist als die in „pfadname“ angegebene, werden gelöscht.

= BEFORE

Alle durch „pfadname“ ausgewählten Generationen, deren Generationsnummer kleiner ist als die in „pfadname“ angegebene, werden gelöscht.

PREFIX = pre

Nur zusammen mit MF=D:

„pre“ ist eine 1-3 Zeichen lange Zeichenfolge, die die entsprechende Zeichenfolge am Beginn der generierten Namen ersetzt und so aufrufspezifische Namen erzeugt. Das erste Zeichen von „pre“ muss ein Buchstabe sein.

PREFORM

Löscht Dateien abhängig von deren (beabsichtigten) Dateiformat auf SM-Pubsets.

= *ANY

Das Dateiformat ist kein Auswahlkriterium.

= *NONE

Löscht alle Dateien, für die kein PREFORM-Wert definiert wurde.

= *K

Löscht alle Dateien, die das beabsichtigte Dateiformat *K besitzen.

= *NK2

Löscht alle Dateien, die das beabsichtigte Dateiformat *NK2 besitzen.

= *NK4

Löscht alle Dateien, die das beabsichtigte Dateiformat *NK4 besitzen.

= (list-of-preform)
Löscht alle Dateien, die einem der angegebenen Dateiformate entsprechen. Innerhalb der Liste können alle Werte außer ANY angegeben werden.

PROTACT
Der Anwender kann die zu bearbeitenden Dateien auswählen abhängig von der Schutzstufe der höchsten aktivierten Zugriffskontrolle.

Für Zugriffe auf die Datei gilt der höchste aktivierte Zugriffsschutz. Die nachfolgende Tabelle zeigt Art der Zugriffskontrolle, Schutzmerkmal, das im CATAL-Makroaufruf anzugeben ist und die Rangfolge (Schutzstufe):

Zugriffsschutz	**Schutzmerkmal**	**Schutzstufe**
Standard-Zugriffskontrolle	ACCESS u. SHARE	0
Einfache Zugriffskontroll-Liste	BASACL, OWNERAR, GROUPAR, OTHERAR	1
Zugriffskontroll-Liste (ACL)	ACL (nur mit SECOS < V4.0A)	2
Zugriffskontrolle über GUARDS	GUARDS	2

ACL und GUARDS schließen sich gegenseitig aus. Alle weiteren Schutzmerkmale der Datei (z.B. Kennwörter) werden unabhängig von der realisierten Schutzstufe ausgewertet.

= ANY
Die zu bearbeitenden Dateien werden unabhängig von der Schutzstufe der höchsten aktivierten Zugriffskontrolle ausgewählt.

= LEVEL-0
Es werden nur Dateien bearbeitet, bei denen die Zugriffe über die Standard-Zugriffskontrolle erfolgen.

= LEVEL-1
Es werden nur Dateien bearbeitet, bei denen die Zugriffe über eine einfache Zugriffskontroll-Liste (BASIC-ACL-Schutz) erfolgen.

= LEVEL-2
Es werden nur Dateien bearbeitet, bei denen die Zugriffe über eine Zugriffskontroll-Liste (ACL) oder über GUARDS erfolgen. ACL-geschützte Dateien können nur noch mit der Angabe IGNORE=*ACCESS gelöscht werden.

= (list-of-protact)
Der Anwender kann in einer Liste maximal 3 Schutzstufen angeben. Es werden nur Dateien bearbeitet, bei denen die Zugriffe über eine Zugriffskontrolle erfolgen, die einer der angegebenen Schutzstufen entspricht.

RELSPAC
Der Anwender kann die zu bearbeitenden Dateien auswählen abhängig von einer vereinbarten Sperre gegen Freigabe von nicht belegtem Speicherplatz mit dem FILE-Makroaufruf bzw. dem Kommando MODIFY-FILE-ATTRIBUTES. Die Sperre kann mit dem CATAL-Makro im Katalog vereinbart werden.

= ANY
Die Sperre gegen Freigabe von nicht belegtem Speicherplatz ist kein Auswahlkriterium.

= ALLOWED
Es werden alle Dateien bearbeitet, bei denen nicht belegter Speicherplatz freigegeben werden darf.

= IGNORED
Es werden alle Dateien bearbeitet, bei denen nicht belegter Speicherplatz nicht freigegeben werden darf.

SHARE
Der Anwender kann die zu bearbeitenden Dateien auswählen abhängig von ihrer Mehrbenutzbarkeit (siehe Operand SHARE im CATAL-Makro).

= ANY
Die Mehrbenutzbarkeit ist kein Auswahlkriterium.

= YES
Es werden nur Dateien bearbeitet, die mehrbenutzbar sind, d.h. Dateien, die bei aktiver Standard-Zugriffskontrolle auch für fremde Benutzerkennungen zugreifbar sind.

= NO
Es werden nur Dateien bearbeitet, die nicht mehrbenutzbar sind, d.h. Dateien, die bei aktiver Standard-Zugriffskontrolle nur für den Dateieigentümer zugreifbar sind.

= SPECIAL
Es werden nur mehrbenutzbare (siehe YES) Dateien bearbeitet, die auch für die Kennungen mit dem Privileg Hardware-Maintenance zugreifbar sind.

= (list-of-share)
In einer Liste können mehrere Operandenwerte angegeben werden.

SIZE

Nur für Plattendateien:

Der Anwender kann über die Dateigröße bzw. Größe des reservierten Speicherplatzes (= Anzahl der PAM-Seiten) bestimmen, welche Dateien vom ERASE bearbeitet werden sollen.

Dateigenerationsgruppen und Dateigenerationen werden nicht berücksichtigt.

Die Operandenwerte „zahl“ geben eine Anzahl PAM-Seiten an; Bereichsangaben gelten jeweils einschließlich der Bereichsgrenzen; $0 \leq$ zahl ≤ 2147483647

= ANY

Die Größe des reservierten Speicherplatzes ist kein Auswahlkriterium.

= FSIZE

Es werden nur Dateien bearbeitet, für die zwar Speicherplatz reserviert ist, die jedoch noch keinen Speicherplatz belegen (LASTPG = 0), d.h. die noch nicht eröffnet wurden.

= zahl

Es werden nur Dateien bearbeitet, für die genau so viele PAM-Seiten reserviert sind wie mit „zahl“ angegeben.

= (zahl[,])

Es werden nur Dateien bearbeitet, für die mindestens so viele PAM-Seiten reserviert wurden wie angegeben (SIZE $\geq$ zahl).

= (,zahl)

Es werden nur Dateien bearbeitet, für die höchstens so viele PAM-Seiten reserviert sind wie angegeben (SIZE $\leq$ zahl).

= (zahl1,zahl2)

Es werden alle Dateien bearbeitet, für die mindestens so viele PAM-Seiten reserviert sind wie mit „zahl1“ angegeben und höchstens so viele wie mit „zahl2“ angegeben.

SLEVEL
Der Anwender kann über die Speicherhierarchie-Ebene die Dateien auswählen, die von ERASE bearbeitet werden sollen. HSMS unterstützt die folgenden Speicherhierarchie-Ebenen:

S0: realisiert durch Plattenspeicher mit schnellem Zugriff (Online-Verarbeitung)
S1: realisiert durch Plattenspeicher mit hoher Kapazität (online-verfügbare Hintergrundebene)
S2: realisiert durch Magnetband- oder Magnetbandkassettenarchive (offline-verfügbare Hintergrundebene)

= ANY
Es werden die angegebenen Dateien bearbeitet, unabhängig von der Speicherhierarchie-Ebene, auf der sie sich befinden.

= S0
Es werden nur Dateien bearbeitet, die sich auf der Ebene S0 befinden.

= S1
Es werden nur Dateien bearbeitet, die sich auf der Ebene S1 befinden.

= S2
Es werden nur Dateien bearbeitet, die sich auf der Ebene S2 befinden.

= (list-of-slevel)
Der Anwender kann in einer Liste maximal 3 Speicherhierarchie-Ebenen angeben. Es werden nur Dateien bearbeitet, die sich auf einer der angegenen Speicherhierarchie-Ebenen befinden.

STATE
Der Anwender kann die zu bearbeitenden Dateien über ihren momentanen Bearbeitungszustand auswählen.

= ANY
Die Speicherhierarchie-Ebene ist kein Auswahlkriterium.

= NOCLOS
Es werden nur Dateien bearbeitet, die momentan schreibend geöffnet sind. Das können sein:
– normal eröffnete Dateien (Openmodus OUTIN, INOUT, OUTPUT)
– in einer vorherigen Session nicht geschlossene Dateien
– in der laufenden Session nicht geschlossene Dateien, weil der Prozess abgebrochen wurde.

= CLOSED
Es werden nur Dateien bearbeitet, die bereits geschlossen wurden, d.h. Dateien, die nicht durch NOCLOS ausgewählt werden.

= CACHED
Es werden nur Dateien bearbeitet, die sich momentan in einem Cache befinden.

= NOT-CACHED
Es werden nur Dateien bearbeitet, die momentan nicht über einen Cache verarbeitet werden.

= CACHE-NOT-SAVED
Es werden nur Dateien bearbeitet, für die es beim Schließen nicht möglich war, alle Schreibdaten vom Cache auf einen Plattenspeicher zu sichern.

= REPAIR-NEEDED
Es werden nur Dateien bearbeitet, die schreibend eröffnet sind, und auf die noch kein VERIFY durchgeführt wurde (siehe VERIFY Makro).

= DEFECT-REPORTED
Es werden nur Dateien bearbeitet, die defekte Plattenblöcke enthalten können.

= NO-OPEN-ALLOWED
Es werden nur Dateien bearbeitet, die wegen Dateninkonsistenz nicht geöffnet werden können.

= OPEN-ALLOWED
Es werden nur Dateien bearbeitet, die geöffnet werden können.

= (list-of-state)
Der Anwender kann in einer Liste maximal 4 Dateizustände angeben. Es werden nur Dateien bearbeitet, die sich in einem der angegebenen Zustände befinden.

STOCLAS
Der Anwender kann die zu bearbeitenden Dateien entsprechend der Storage-Klasse zur Dateiablage auf SM-Pubsets auswählen.

= *ANY
Die Storage-Klasse ist kein Auswahlkriterium.

= *NONE
Nur Dateien, für die keine Storage-Klasse definiert ist, werden ausgewählt.

= <c-string 1..8>
Nur Dateien mit der angegebenen Storage-Klasse werden ausgewählt.

STOTYPE
Der Anwender kann die zu bearbeitenden Dateien entsprechend des Speichertyps auswählen .

= *ANY
Der Speichertyp ist kein Auswahlkriterium.

= *PUBSPACE
Nur Dateien, die auf gemeinschaftlichen Datenträgern liegen, werden ausgewählt.

= *NETSTOR
Nur Dateien, die auf Net-Storage-Volumes liegen, werden ausgewählt.

SUPPORT
Der Anwender kann über den Datenträgertyp festlegen, welche Dateien vom ERASE bearbeitet werden sollen.
Dateigenerationsgruppen und Dateigenerationen werden nicht berücksichtigt.

= ANY
Der Datenträgertyp ist kein Auswahlkriterium.

= PUBLIC
Es werden nur Dateien auf Public-Platten und auf Net-Storage bearbeitet.

= PRDISC
Es werden nur Dateien auf Privatplatten bearbeitet.

= TAPE
Es werden nur Dateien auf Magnetband oder Magnetbandkassette bearbeitet.

= (list-of-support)
Der Anwender kann in einer Liste maximal 3 Datenträgertypen angeben. Es werden nur Dateien bearbeitet, die auf einem der angegebenen Datenträgertypen abgespeichert sind.

S0MIGR
Bearbeitet werden Dateien, abhängig davon, ob eine Umallokierung (Migration) auf S0-Ebene erlaubt ist.

= *ANY
Die Migrations-Erlaubnis ist kein Auswahlkriterium.

= *ALLOWED
Bearbeitet werden nur Dateien, für die eine Migration innerhalb der S0-Ebene erlaubt ist.

= *FORBIDDEN
Bearbeitet werden nur Dateien, für die eine Migration innerhalb der S0-Ebene nicht erlaubt ist.

= (list-of-s0migr)
Der Anwender kann die gewünschten Werte in einer Liste angeben. Bearbeitet werden alle Dateien, für die im Katalog einer der angegebenen Werte vereinbart wurde.

TIMBASE
Steuert, ob die absoluten Datumseingaben in UTC- oder lokaler Zeit erfolgen. Dies betrifft die Operanden CRDATE, DELDATE, EXDATE, LADATE und LCDATE. Relative Datumsangaben beziehen sich stets auf die lokale Zeit.

= *UTC
Absolute Datumseingaben erfolgen in UTC-Zeit.

= *LTI
Absolute Datumseingaben erfolgen in lokaler Zeit.

TYPE
Der Anwender kann über den Dateityp Dateien auswählen, die bearbeitet werden sollen. Von TYPE ist auch abhängig, welche Auswahlkriterien bei der ERASE-Bearbeitung ausgewertet werden, da Dateigenerationsgruppen und Dateigenerationen nicht von allen Selektionsparametern berücksichtigt werden.

= ANY
Es werden sowohl „normale“ Dateien als auch Dateigenerationsgruppen und Dateigenerationen vom ERASE bearbeitet. Allerdings werden FGG und Generationen nicht bei allen Selektionsparametern berücksichtigt, damit keine Lücken in der Generationenfolge entstehen.

= FILE
Es werden keine Dateigenerationsgruppen oder Dateigenerationen vom ERASE bearbeitet, alle weiteren Selektionsoperanden werden ausgewertet.

= FGG
Es werden nur Dateigenerationsgruppen und Dateigenerationen vom ERASE bearbeitet. In Verbindung mit TYPE=FGG sind nur solche Selektionsoperanden sinnvoll, die sich auf Merkmale beziehen, die für alle Generationen einer FGG gleich sind: ACCESS, ACL, BACKUP, CCS, DELDATE, EXDATE, MANCLAS, MIGRATE, PASS, RELSPAC, SHARE, SUPPORT=PRDISC und WORKFIL.

Dateigenerationsgruppen bzw. Dateigenerationen werden nicht zur Bearbeitung ausgewählt:

– wenn der Selektionsoperanden VOLUME nicht zusammen mit CATALOG angegeben ist oder keine Privatplatte bezeichnet.
– wenn ein Selektionsoperand angegeben wird, der ein Merkmal bezeichnet, dass nicht für alle Generationen/FGG gleich ist.

= PLAM
Es werden nur PLAM-Bibliotheken bearbeitet. Dies ist eine Untermenge der Dateien, die bei der Angabe TYPE=FILE ausgewählt werden.

= (list-of-type)
Der Anwender kann in einer Liste maximal 3 Dateitypen (FILE, FGG, PLAM) angeben. Es werden nur Dateien bearbeitet, die einem der angegebenen Dateitypen entsprechen.

USRINFO
Der Anwender kann abhängig von der benutzereigenen Metainformation Dateien/Dateigenerationen zur Bearbeitung auswählen.

= *ANY
Die benutzereigene Metainformation ist kein Auswahlkriterium.

= *NONE
Es werden nur Dateien bearbeitet, die keine benutzereigene Metainformation besitzen.

= <c-string 1..8>
Es werden nur Dateien mit der angegebenen benutzereigenen Metainformation bearbeitet.

VERSION
Gibt an, welche Version der Parameterliste generiert werden soll.

= 0
Voreinstellung: Es wird das Parameterlistenformat generiert, das vor BS2000 V9.5A unterstützt wurde.
Dieses Format unterstützt allerdings auch nur die bis dahin bekannten Parameter. Z.B. darf der Pfadname nur ohne Musterzeichen angegeben werden und von den Selektionsparametern ist nur VOLUME und POS erlaubt. Die unterstützten Operanden/Operandenwerte können der Tabelle „Versionsunterschiede VERSION=0/1/2“ auf Seite 390 entnommen werden.

= 1
Es wird das Parameterlistenformat generiert, das in BS2000 V9.5 und V10.0 unterstützt wurde.
Dieses Format unterstützt allerdings auch nur die bis dahin bekannten Parameter. Die unterstützten Operanden/Operandenwerte können der Tabelle „Versionsunterschiede VERSION=0/1/2“ auf Seite 390 entnommen werden.

= 2
Es wird das Parameterlistenformat für die Version ab BS2000/OSD-BC V1.0 generiert.

= 3
Es wird das Parameterlistenformat für die Version ab BS2000/OSD-BC V3.0 generiert.

Hinweis
Wenn schon bestehende Software neu übersetzt werden soll, die Manipulationen an der generierten Parameterliste vornimmt, muss das alte Format angefordert werden. Ansonsten liegt Source-Kompatibilität vor.

VOLSET
Der Anwender kann die zu bearbeitenden Dateien über den Volume-Set auswählen, auf dem sie liegen.

= *ANY
Der Volume-Set ist kein Auswahlkriterium.

= <c-string 1..4>
Ausgewählt werden alle Dateien, die auf dem spezifizierten Volume-Set liegen.

VOLUME
Der Anwender kann die zu bearbeitenden Dateien über die Archivnummer (VSN) ihres Datenträgers auswählen.

= *ANY
Es werden alle Dateien bearbeitet unabhängig von der Archivnummer ihrer Datenträger.

= vsn
Es werden alle Dateien bearbeitet, die Speicherplatz auf dem angegebenen Datenträger belegen. Bei gleichzeitiger Angabe der Aktionsoperanden DESTROY, SPACE, SPACE-CATALOG bzw. DATA werden keine Dateigenerationen und Dateigenerationsgruppen ausgewählt. Bei gleichzeitiger Angabe des Aktionsoperanden CATALOG werden keine Dateigenerationen auf Magnetbändern ausgewählt.

WORKFIL
Der Anwender kann die zu bearbeitenden Dateien auf SM-Pubsets, abhängig davon auswählen, ob sie vom Systemverwalter gelöscht werden können (Arbeitsdateien).

= *ANY
Es ist kein Auswahlkriterium, ob die Dateien Arbeitsdateien sind oder nicht.

= *NO
Bearbeitet werden alle Dateien, die keine Arbeitsdateien sind.

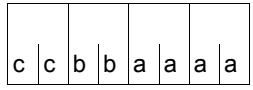
= *YES
Bearbeitet werden alle Dateien, die Arbeitsdateien sind.

Hinweise zur Programmierung

1. Der Fehlercode wird nur noch im Standardheader der Parameterliste und nicht mehr wie bis VERSION=2 im Register 15 hinterlegt.
2. Fehlerschlüssel 06D6 – „dateiname“ teilqualifiziert und es konnten nicht alle Dateien gelöscht werden
3. Fehlerschlüssel 05DF – *SYSOUT im Dialogbetrieb nicht zulässig
4. In bestimmten Fehlerfällen (Parameterbereich nicht zugreifbar oder nicht ausgerichtet) wird eine Programm-Terminierung mit STXIT-Anschluss eingeleitet.

Returncodes

Standard-header:



Über die Ausführung des Makros ERASE wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Kein Fehler
	X'40'	X'0501'	Angeforderter Katalog nicht verfügbar
	X'82'	X'0502'	Angeforderter Katalog im Ruhezustand
	X'40'	X'0503'	Falsche Information im MRSCAT
	X'82'	X'0504'	Fehler im Katalogverwaltungssystem
	X'40'	X'0505'	Fehler bei der Rechner-Kommunikation (MRS)
	X'80'	X'0506'	Operation wegen Masterwechsel abgebrochen
	X'40'	X'0510'	Fehler beim Aufruf einer internen Funktion
	X'40'	X'0512'	Angeforderter Katalog unbekannt
	X'40'	X'051A'	Datei existiert bereits
	X'40'	X'051B'	Benutzerkennung im angegebenen Pubset unbekannt
	X'40'	X'051C'	Kein Zugriffsrecht auf angegebenen Pubset
	X'40'	X'051D'	LOGON-Passwort auf angegebenem Pubset anders
	X'20'	X'0530'	Fehler bei Speicherplatz-Anforderung
	X'20'	X'0531'	Unerwarteter Fehler beim Katalogzugriff
	X'82'	X'0532'	Datei in Gebrauch und damit gesperrt
	X'40'	X'0533'	Angegebene Datei nicht gefunden
	X'82'	X'0534'	Privater Datenträger kann nicht zugewiesen werden
	X'40'	X'0535'	Keine Zugriffsberechtigung auf den Katalogeintrag der Datei
	X'20'	X'053B'	Systemfehler beim Dateizugriff
	X'40'	X'053D'	Katalog oder F1-Etikett-Block ist zerstört
	X'82'	X'053F'	Datei ist von einer anderen Task reserviert
	X'20'	X'054F'	Unerwarteter Fehler beim Zugriff auf JOIN-Datei
	X'40'	X'055C'	Katalogeintrag auf Privatplatte nicht gefunden
	X'01'	X'0571'	Systemdatei als *DUMMY erklärt
	X'40'	X'0572'	Systemdatei ist nicht einer DVS-Datei zugewiesen
	X'40'	X'0574'	DVS-Fehler beim Löschen einer Systemdatei
	X'82'	X'0575'	Systemkommando für diese Systemdatei aktiv

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'01'	X'0576'	Widersprüchliche Operandenkombination oder reservierte Felder des Parameterbereiches verwendet
	X'20'	X'0577'	Interner Fehler beim Zugriff auf die Auftragsumgebung
	X'20'	X'0578'	Interner Fehler bei Überprüfung der Zugriffsrechte
	X'40'	X'057C'	HSMS hat Recall abgewiesen
	X'40'	X'057D'	HSMS-Datei verdrängt. Zurückholen ohne Verzögerung nicht möglich
	X'40'	X'057E'	HSMS nicht verfügbar
	X'82'	X'0594'	Nicht genug virtueller Speicher verfügbar. Dieser Returncode kann insbesondere auch im Zusammenhang mit einer Auswahlangabe (Wildcard) auftreten, wenn zu viele Dateien selektiert werden
	X'01'	X'0599'	Operand wird in der RFA-BS-Version nicht unterstützt
	X'01'	X'05AB'	Adresse des Ausgabebereiches falsch oder nicht angegeben
	X'01'	X'05AC'	Fehlerhafter zweiter Operand
	X'40'	X'05B3'	Angabe einer fremden Benutzerkennung nur dem Systemverwalter erlaubt
	X'40'	X'05BF'	Datei mit Kennwort geschützt
	X'82'	X'05C3'	Zu löschende Dateigeneration gesperrt
	X'01'	X'05C5'	SPACE-Angabe für Dateien auf Privatplatten nicht erlaubt
	X'40'	X'05C6'	Freigabedatum erlaubt das Löschen der Datei nicht
	X'20'	X'05C7'	Interner Fehler im DMS
	X'01'	X'05C9'	Nur Dateien auf privaten Datenträgern können exportiert werden
	X'82'	X'05D0'	Dateigesperrt weil in Gebrauch
	X'01'	X'05DE'	Dateiname nicht vorhanden oder unzulässig
	X'01'	X'05EE'	Dateiname zu lang
X'02'	X'00'	X'05F7'	Dateigeneration existiert nicht aber Gruppeneintrag wird geändert
	X'01'	X'05FA'	Zugriff auf REMOTE-IMPORTED Pubset nicht möglich
	X'40'	X'05FC'	Angegebene Benutzerkennung nicht im HOME-Pubset
	X'40'	X'0609'	Aktion für Systemdatei nicht erlaubt
	X'40'	X'0640'	Zugriff auf Net-Storage wird vom Subsystem ONETSTOR wegen Kommunikationsproblemen mit dem Net-Client abgewiesen
	X'40'	X'0643'	Net-Client meldet Zugriffsfehler
	X'40'	X'0644'	Net-Client meldet internen Fehler
	X'40'	X'0645'	Datei auf Net-Storage nicht vorhanden
	X'40'	X'0649'	Net-Server meldet POSIX-ACL-Fehler

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'064A'	Net-Client meldet, dass der Zugriff auf Dateien auf dem Net-Storage-Volume verboten ist
	X'40'	X'064B'	Zugriff auf Node-Files vom Net-Client nicht unterstützt
	X'40'	X'064C'	Verzeichnis der angegebenen Benutzerkennung existiert nicht auf Net-Server
	X'40'	X'0666'	Die Datei ist durch ACL oder GUARDS schreibgeschützt
	X'20'	X'069D'	Fehlerhaft aufgebauter Katalogeintrag
X'01'	X'00'	X'06B4'	Vor oder nach der angegebenen Generation existiert keine zu löschende Generation
	X'01'	X'06C7'	Ungültige Generationsnummer angegeben
	X'40'	X'06CC'	nur bei Auswahlangabe (Wildcard): keine Datei entspricht der Auswahlangabe
	X'01'	X'06D4'	Unzulässige Generationsangabe
	X'40'	X'06D5'	Datei geschützt
X'02'	X'00'	X'06D6'	Fehler beim Löschen einiger Dateien
	X'01'	X'06F5'	Keine Berechtigung zur Nutzung der angegebenen Operanden (TPR oder TSOS erforderlich)
	X'01'	X'06F9'	Entweder muss der Dateiname oder Volume angegeben werden
	X'01'	X'06FD'	Parameterbereich ungültig oder nicht zugreifbar
	X'40'	X'06FF'	BCAM-Verbindung abgebrochen
	X'01'	X'FFFF'	Falsche Funktionsnummer im Parameterbereichs-Header
	X'03'	X'FFFF'	Falsche Versionsnummer im Parameterbereichs-Header

Versionsunterschiede VERSION=0/1/2

Die folgende Tabelle „Versionsunterschiede“ zeigt, welche Operanden/Operandenwerte mit VERSION=2/1/0 unterstützt werden.

Im Makroaufrufformat mit VERSION=2 können alle Operanden/Operandenwerte verwendet werden, die bis einschließlich der Version BS2000/OSD-BC V2.0A unterstützt wurden.

Im Makroaufrufformat mit VERSION=1 können alle Operanden/Operandenwerte verwendet werden, die bis einschließlich der BS2000-Version V10.0A unterstützt wurden.

Im Makroaufrufformat mit VERSION=0 können alle Operanden/Operandenwerte verwendet werden, die bis einschließlich der BS2000-Version V9.0A unterstützt wurden.

Operand	Vers=0	Vers=1	Vers=2	Bemerkungen
MF=E	x	x	x	
VERSION	x	x	x	
MF=D	-	x	x	
PREFIX	-	x	x	
VERSION	-	x	x	
MF=L	x	x	x	
*	x	x	x	
*SYSid	x	x	x	
*DUMMY	-	x	x	
CATALOG	x	x	x	
DATA	x	x	x	
DATA-KEEP-ATTR	-	-	-	
DELETE-OR-EXPORT	-	x	x	
DESTROY	x	x	x	
pfadname	x	x	x	*Vers=0:* Keine Musterzeichen erlaubt.
prefix	-	x	x	
SPACE	x	x	x	
SPACE-CATALOG	-	x	x	
ACCCNT	-	-	x	
ACCESS	-	x	x	
ACL	-	-	x	
ADMINFO	-	-	-	

Operand	Vers=0	Vers=1	Vers=2	Bemerkungen	
MF=L (Fortsetzung)					
AVAIL	-	-	-		
BACKUP	-	x	x	*Vers=1:*	Operandenwert (list-of-backup) nicht möglich
BASACL	-	-	x		
BLKCNT	-	-	x		
BLKCTRL	-	x	x	*Vers=1:*	Operandenwerte ANY, DATA4K, DATA2K, NK4, NK2, NK und (list-of-blkctrl) nicht möglich
CCS	-	-	x		
CHECK	-	x	x		
CRDATE	-	x	x	*Vers=1:*	Operandenwerte mit der Angabe (zeit) nicht möglich, siehe [1)]
DELDATE	-	-	-		
DISKWR	-	-	x		
ENCRYPT	-	-	-		
EXDATE	-	x	x	*Vers=1:*	Operandenwerte mit der Angabe (zeit) nicht möglich, siehe [1)]
EXTENTS	-	x	x		
FCBTYPE	-	x	x	*Vers=1:*	Operandenwert (list-of-fcbtype) nicht möglich
FILTYPE	-	-	-		
FSIZE	-	x	x		
GROUPAR	-	-	x		
GUARDS	-	-	x		
IGNORE	-	x	x		
IOPERF	-	-	x		
IOUSAGE	-	-	x		
KEEPACL	-	-	x		
LADATE	-	x	x	*Vers=1:*	Operandenwerte mit der Angabe (zeit) nicht möglich, siehe [1)]
LASTPAG	-	-	x		
LCDATE	-	-	x		
LIST	-	x	x		
MANCLAS	-	-	-		
MIGRATE	-	x	x	*Vers=1:* *Vers=0/1/2:*	Operandenwert (list-of-migrate) nicht möglich Operandenwert FORBIDDEN nicht möglich

Operand	Vers=0	Vers=1	Vers=2	Bemerkungen
MF=L (Fortsetzung)				
MOUNT	-	x	x	
NOSTEP	-	x	x	
OTHERAR	-	-	x	
OWNERAR	-	-	x	
PASS	-	x	x	*Vers=1:* Operandenwert (list-of-pass) nicht möglich
PASSWD	-	x	x	
POS	x	x	x	
PREFIX	-	x	x	
PROTACT	-	-	x	
RELSPAC	-	-	x	
SHARE	-	x	x	*Vers=1:* Operandenwert (list-of-share) nicht möglich
SIZE	-	x	x	
SLEVEL	-	x	x	
STATE	-	-	x	*Vers=2*: Operandenwerte CACHE-NOT-SAVED und DEFECT-REPORTED nicht möglich
STOCLAS	-	-	-	
SUPPORT	-	x	x	*Vers=1:* Operandenwert (list-of-support) nicht möglich
S0MIGR	-	-	-	
TIMBASE	-	-	-	
TYPE	-	-	x	
USRINFO	-	-	-	
VERSION	x	x	x	
VOLSET	-	-	-	
VOLUME	x	x	x	*Vers=0:* Nur der Operandenwert vsn möglich *Vers=1:* Nur der Operandenwert vsn möglich
WORKFIL	-	-	-	

Legende

x Operand ist in der Makroversion verfügbar

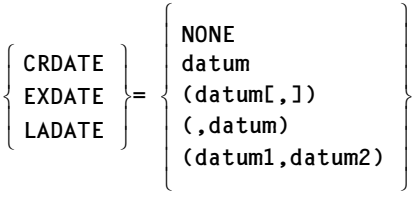
- Operand ist in der Makroversion nicht verfügbar

Vers Version

In der Tabelle sind unter MF=L die Stellungsoperanden vor den Schlüsselwortoperanden eingeordnet.

Hinweis

[1]) Die Operanden CRDATE, EXDATE und LADATE haben in der Makroversion 1 folgendes Format:



EXLST – Exit-Adressenliste anlegen

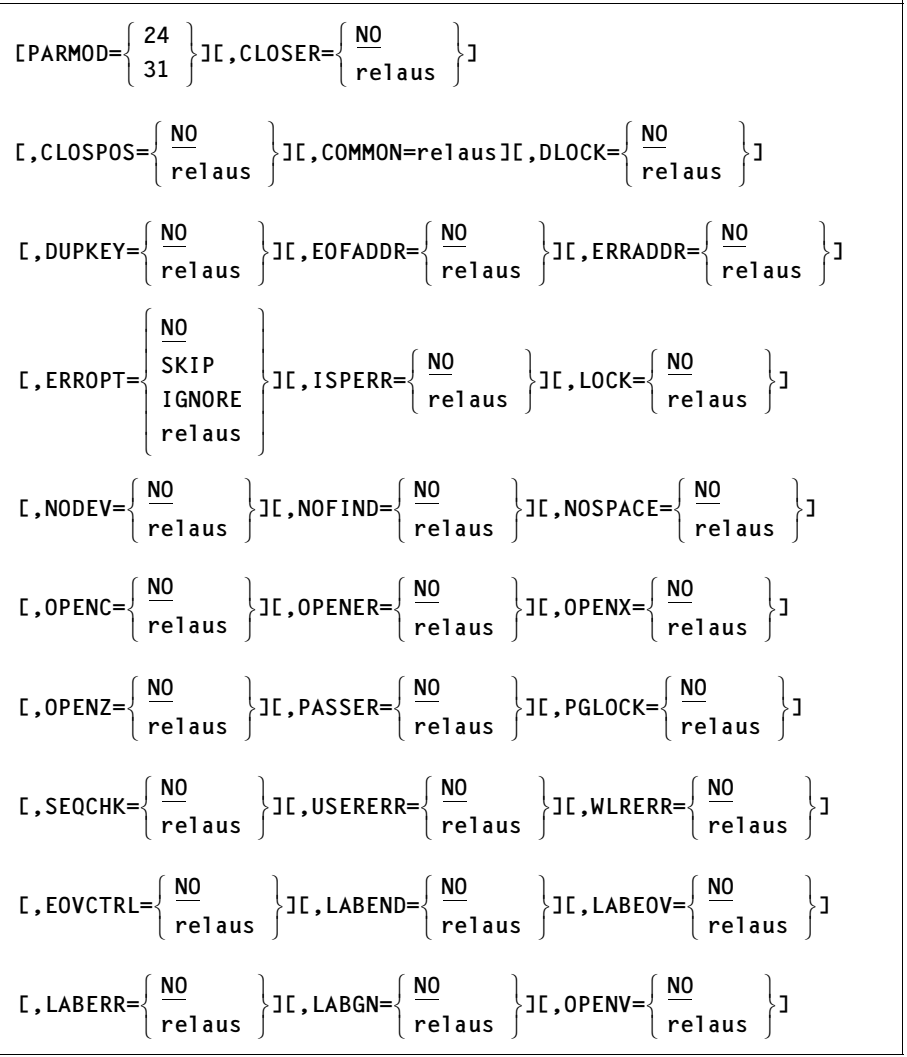
Makrotyp: 0-Typ

Mit dem EXLST-Makroaufruf erstellt der Anwender eine Liste von symbolischen Adressen, die auf Programmroutinen verweisen, in denen Ereignisse, die zur Unterbrechung der normalen Verarbeitung führten, ausgewertet und bearbeitet werden können.

Für jedes Ereignis gibt es einen speziellen Operanden im EXLST-Makro. Ist zu diesem Operanden eine symbolische Adresse im Makroaufruf angegeben, kann das DVS bei Auftreten des Ereignisses an die entsprechende Programmroutine verzweigen. Wird der Operand nicht oder nur mit einem Nullstring versorgt, führt das Auftreten des Ereignisses zum Abbruch – es sei denn, über den Operanden COMMON wurde eine allgemein gültige Fehlerbehandlungs-Routine adressiert.

Die trifft jedoch nicht für die Operanden zu, mit denen bei der OPEN-Verarbeitung der FCB verändert werden kann oder mit denen der Benutzer bei OPEN- oder CLOSE-Verarbeitung eigene Bandkennsätze schreibt. Sind diese Operanden im EXLST-Makroaufruf genannt, verzweigt das DVS automatisch zu den so adressierten Programmroutinen.

Format

Operation	Operanden
EXLST	

Operandenbeschreibung

relaus
symbolische Adresse im Assembler-Programm

PARMOD
Gibt den für den FCB geltenden Generierungsmodus für den Makroaufruf an.

Voreinstellung: der durch den Makro GPARMOD oder den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst. Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig, d.h. im 16-MB-Adressraum.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert, das Objekt ist im 2-GB-Adressraum ablauffähig.

CLOSER
Bei der CLOSE-Verarbeitung trat ein Fehler auf. Der Fehlerschlüssel im FCB-Feld ID1ECB beschreibt den Zustand genauer.

CLOSPOS
Für Band-Eingabedateien mit Nichtstandardkennsätzen:
der Anwender kann ein Band, für das ein CLOSE-Makro mit REPOS oder LEAVE aufgerufen wurde, mit BTAM-Makroaufrufen positionieren. Die CLOSE-Verarbeitung wird nach Beendigung der CLOSPOS-Routine fortgesetzt. Wird der CLOSPOS-Ausgang nicht definiert, führt das System die Positionierung durch.

COMMON
Die Steuerung geht an diese Adresse, wenn für den aufgetretenen Fehler kein spezieller Ausgang angegeben wurde.

Ausnahmen:

CLOSPOS	LABEOV	OPENC	OPENZ
EOVCTRL	LABERR	OPENV	WLRERR
LABEND	LABGN	OPENX	

Wenn die Verarbeitung durch ein Ereignis unterbrochen wird, für das kein EXLST-Operand angegeben wurde, geht die Steuerung an die mit COMMON adressierte Routine. Ausnahmen bilden die o.g. Operanden, die einen Eingriff in die Verarbeitung „erwarten“.

Siehe auch Tabellen „Fehlerausgänge“ am Schluss der EXLST-Beschreibung.

DLOCK
Nur für UPAM-Verarbeitung von Plattendateien:
Deadlock: für einen Auftrag, der bereits Sperren hält, wurden weitere Sperren angefordert, die nicht verfügbar sind.

Der Deadlock-Ausgang wird nach der im FCB-Operanden PAMTOUT angegebenen Wartezeit angesprungen. In der DLOCK-Routine müssen zunächst die „alten“ Sperren freigegeben werden, bevor erneut Sperren angefordert werden können – andernfalls wird das Programm abgebrochen.

DUPEKY
Nur für ISAM:
es wurde versucht, einen Satz in die ISAM-Datei zu schreiben, dessen Schlüssel bereits vorhanden ist, und zwar
– mit dem Makro INSRT, der auch zusammen mit DUPEKY=YES in FILE oder FCB nicht für das Schreiben von Sätzen mit gleichen Schlüsseln verwendet werden kann, oder
– mit dem Makroaufruf PUT, ohne dass vorher DUPEKY=YES definiert wurde.

EOFADDR
Nur für Eingabedateien:
beim Versuch, einen Satz zu lesen, wurde das Dateiende erreicht.

Nach Prüfung aller Kennsätze aktiviert das DVS die EOFADDR-Routine, in der der Anwender die Datei schließen kann.

EOVCTRL
Um nach dem Bereitstellen eines Folgebandes die Kontrolle vom System an den Benutzer zu übertragen, ist dieser Ausgang anzugeben.

Für Eingabedateien:
wird nur bei Mehrspuldateien für jede Folgespule aktiviert, nachdem das System die Standard-Anfangskennsätze des Folgebandes und der Benutzer – falls erforderlich – im LABGN-Ausgang seine Kennsätze geprüft hat.

Bei Ausgabedateien:
Dieser Ausgang wird nur bei Mehrbanddateien und da auch nur bei den Folgebändern aktiviert, nachdem das System das Prüfen und Schreiben der Standard-Anfangskennsätze (VOL, HDR) beendet hat.

Dieser Ausgang ist mit dem Makro EXRTN zu verlassen.

ERRADDR
Während der Dateiverarbeitung trat ein Hardwarefehler auf oder eine Ein-/ Ausgabe wurde abnormal beendet. Status-Byte, Standard-Geräte-Byte, Ablaufteil-Markierungsbyte sowie die 3 Fehler-Bytes werden im FCB gespeichert. Bei ISAM-Verarbeitung können ISAM-Blöcke inkonsistent sein (SAM verwendet die Ausgänge: EOFADDR, ERROPT, USERERR)

ERROPT
Für SAM-Eingabedateien:
beim Lesen eines Blocks trat ein Parity-Fehler auf: das DVS führt mehrere Leseversuche durch, bevor es den Block als fehlerhaft ausweist und versucht, eine ERROPT-Routine zu aktivieren. Ist kein ERROPT-Fehlerausgang vorgesehen, wird das Programm beendet.

Im EXLST-Makroaufruf kann der Programmierer bestimmen. ob er einen Lesefehler ignorieren will, den fehlerhaften Block übergehen will oder eine Fehlerbehandlung in einer ERROPT-Routine durchführen will.

Für SAM-Ausgabedateien:
ERROPT ist nur dann relevant, wenn kein WLRERR-Ausgang definiert wurde und Sätze falscher Länge auftreten.

= SKIP
Der fehlerhafte Block soll übergangen werden, d.h.: es werden dem Programm keine Sätze dieses Blocks bereitgestellt. Der nächste Block wird gelesen, und die Verarbeitung wird mit dem ersten Satz dieses Folgeblocks fortgesetzt.

= IGNORE
Der Fehler soll ignoriert werden, die Sätze des fehlerhaften Blocks werden dem Programm zur Verarbeitung zur Verfügung gestellt.

= relaus
symbolische Adresse einer Anwenderroutine – in dieser Routine darf kein GET-Makro aufgerufen werden für die Datei, zu der der fehlerhafte Block gehört. Die Routine kann das Register 0 auswerten, das die Adresse des fehlerhaften Blocks enthält und ein Kennzeichen für die weitere Verarbeitung setzen (im Register 0). Die Verarbeitung wird mit dem Makroaufruf EXRTN fortgesetzt.

Enthält das Mehrzweckregister 0 den Wert X'00000001', wird der aktuelle Block übergangen (≙ SKIP) und die Verarbeitung ab dem nächsten Block fortgesetzt. Jede andere Anzeige bedeutet, dass der Block so verarbeitet werden soll, als ob kein Fehler aufgetreten wäre.

ISPERR
Für ISAM-Dateien mit Index- und Datenteil auf verschiedenen privaten Platten:
Für die Erweiterung des Indexteils steht nicht genügend Platz zur Verfügung (bei diesen ISAM-Dateien kann der Speicherplatz separat für den Index- oder für den Datenteil erweitert werden (vgl. FILE-Makro)).

LABEND

Für Banddateien:
Zum Prüfen/Erstellen von Benutzer-Dateiendekennsätzen; wird dieser Ausgang nicht verwendet, ignoriert das System alle Benutzerkennsätze.

Bei Eingabedateien:
Nach dem Erkennen des Dateiendes aktiviert das System den LABEND-Ausgang, bevor der EOFADDR-Ausgang aktiviert wird. Der Benutzer kann in dieser Routine seine Dateiendekennsätze (UTL) prüfen. Das System stellt ihm im Register 0 die Adresse eines UTL zur Verfügung. Für Eingabedateien mit Nichtstandardkennsätzen erhält der Benutzer über den LABEND-Ausgang die Möglichkeit, seine Kennsätze, falls vorhanden, bei Dateiende einzulesen und zu prüfen.

Bei Ausgabedateien:
Der LABEND-Ausgang wird nach dem Schließen der Datei durch den Benutzer und dem Schreiben der Benutzer-Endekennsätze (UTL) aktiviert. Das System stellt in Register 0 eine Adresse zur Verfügung, an der der Benutzer seine UTL-Kennsätze bereitstellen muss. Für Ausgabedateien mit Nichtstandardkennsätzen erhält der Benutzer, nachdem er den CLOSE-Makroaufruf angestoßen hat, hier die Kontrolle zum Schreiben der Nichtstandardkennsätze.

Nachdem die Kontrolle wieder an das System zurückgegeben wurde (LBRET-Makroaufruf), wird die CLOSE-Verarbeitung abgeschlossen.

LABEOV

Für Banddateien:
Zum Prüfen/Erstellen von Benutzer-Bandendekennsätzen.

Bei Eingabedateien:
wird der LABEOV-Ausgang aktiviert, nachdem das Bandende erkannt und die Endekennsätze geschrieben wurden. Hier kann der Benutzer die Benutzerkennsätze (UTL) prüfen. Im Register 0 wird ihm vom System die Adresse eines UTL zur Verfügung gestellt. Bei Eingabedateien mit Nichtstandardkennsätzen kann der Benutzer über diesen Ausgang seine Kennsätze hinter seiner Datei, falls vorhanden, einlesen und prüfen.

Bei Ausgabedateien:
wird dieser Ausgang aktiviert, wenn das System Bandende erkannt hat (oder ein FEOV-Makro aufgerufen wurde) und die Bandendekennsätze (EOV) geschrieben sind.
Das System stellt in Register 0 eine Adresse zur Verfügung, an der der Benutzer seine Endekennsätze (UTL) bereitstellen muss. Für Ausgabedateien mit Nichtstandardkennsätzen erhält der Benutzer über diesen Ausgang die Kontrolle vom System entweder nach Erkennen von Bandende, oder wenn er einen FEOV-Makroaufruf abgesetzt hat, um seine Kennsätze zu schreiben.

Bei BTAM-Verarbeitung muss der Benutzer nach Erkennen von Bandende im Feld ERRBYTE einen FEOV-Makro aufrufen. SAM leitet automatisch die EOV-Bearbeitung ein. Nachdem die Kontrolle dem System zurückgegeben wurde (LBRET-Makroaufruf), kann der Bandwechsel durchgeführt werden.

LABERR

Für Banddateien:
für Dateien mit Standardkennsätzen verzweigt das System zu diesem Ausgang, wenn während der Bandende-Verarbeitung ein Fehler aufgetreten ist. Folgender Fehler-Code wird in das ID1ECB-Feld des FCB übertragen:

X'0DE9' Anstatt der erwarteten EOV-/EOF-Kennsätze wurde eine Bandmarke gelesen.

X'0DEA' Es wurde nicht der erwartete EOV-/EOF-Kennsatz gelesen.

X'0DEB' Bandende (Doppel-Abschnittsmarke) wurde erkannt, ohne dass die erwarteten EOV/EOF-Kennsätze gelesen wurden.

X'0DEC' Prüfung des Blockzählers war negativ.

Der Benutzer muss die Kontrolle mit dem EXRTN-Makroaufruf wieder an das System zurückgeben.

Das Register 0 muss einen der folgenden Werte enthalten, um dem System mitzuteilen, wie es weiterarbeiten soll:

X'00' Bandwechsel durchführen, als ob die richtigen EOV/EOF-Kennsätze gelesen worden wären

X'01' Prozess mit Fehler beenden

X'02' Dateiende-Verarbeitung durchführen

Wird der LABERR-Ausgang nicht verwendet und tritt einer der oben beschriebenen Zustände auf, so gibt das System dem Benutzer eine Fehlermeldung. Er kann entweder das Programm fortsetzen oder eine CLOSE-Routine anstoßen.

Wird dieser Ausgang nicht angegeben, ignoriert das System Benutzerkennsätze bei Eingabedateien, und es können für Ausgabedateien keine erstellt werden!

LABGN
Für Banddateien:

Bei Eingabedateien:
wird der LABGN-Ausgang nach Prüfung der Standardanfangskennsätze (VOL, HDR) aktiviert; in der LABGN-Routine können die Benutzer-Anfangskennsätze (UHL) geprüft werden. Die Adresse des Kennsatzes im Puffer wird im Register 0 an das Benutzerprogramm übergeben. Enthält die Datei Nichtstandardkennsätze, kann der Benutzer in der LABGN-Routine seine Kennsätze einlesen und prüfen.

Bei Ausgabedateien, nach OPEN-Verarbeitung, und nach Prüfen und Schreiben der Standard-Anfangskennsätze:
Im Register 0 stellt das System dem Benutzer eine Adresse zur Verfügung, an der er seine Kennsätze (UHL) bereitstellen muss. Soll die Datei Nichtstandardkennsätze erhalten, können sie in der LABGN-Routine erstellt werden.

Wird die Kontrolle wieder an das System zurückgegeben, (LBRET-Makroaufruf), muss das Band bei OPEN EXTEND hinter den letzten Datenblock bzw. bei OPEN REVERSE vor den ersten Datenblock positioniert sein. Gilt TPMARK = YES, wird das Band bei OPEN INPUT um eine Abschnittsmarke zurückgespult, bei OPEN REVERSE um eine Abschnittsmarke vorgespult. Bei TPMARK=NO setzt das System voraus, dass das Band vor den ersten Datenblock positioniert ist.

LOCK
Die Datei ist gesperrt; sie kann nicht eröffnet werden, weil sie bereits von einem anderen Auftrag eröffnet wurde und die OPEN-Modi unverträglich sind: mindestens einer der Aufträge hat einen OPEN-Modus ≠ INPUT gewählt.

NODEV
Es ist kein Gerät frei, auf dem der private Datenträger bereitgestellt werden kann, oder der private Datenträger wird z.Z. von einem anderen Anwender genutzt (vorherige Gerätereservierung empfohlen!).

NOFIND
Für ISAM:
die Aktionmakroaufrufe GETKY, ELIM (mit KEY-Angabe) oder GETFL konnten nicht erfolgreich ausgeführt werden:
– GETKY / ELIM – die Datei enthält keinen Satz mit dem angegebenen Schlüssel
– GETFL – der definierte Dateibereich enthält keinen Satz, der die Flag-Bedingung erfüllt.

NOSPACE
Für Plattendateien:
der benötigte Speicherplatz kann nicht bereitgestellt werden.

OPENC
Die Datei war bei einer vorhergehenden Bearbeitung als Ausgabedatei eröffnet und wurde nicht ordnungsgemäß geschlossen. In der OPENC-Routine kann der VERIF-Makro genutzt werden, um die Datei zu schließen und die Konsistenz wiederherzustellen. (Das Feld ID1ECB des FCB enthält den DVS-Fehlercode X'0DD1'). Ohne OPENC-Ausgang setzt das System die OPEN-Verarbeitung fort.

OPENER
Fehler beim Eröffnen der Datei: z.B. Widerspruch im FCB, der Datei wurde kein Speicherplatz zugewiesen. Ein Fehlerschlüssel, der den Zustand näher beschreibt, wird im FCB gespeichert.

OPENV
Für Banddateien:

Für Dateien mit Standardkennsätzen:
In der OPENV-Routine kann der Benutzer die UVL-Kennsätze prüfen (bei Eingabedateien) bzw. schreiben (für Ausgabedateien). Im Register 0 stellt das DVS die Adresse zur Verfügung, an der bei Eingabedateien der Kennsatz zu finden ist oder an der der Benutzer bei Ausgabedateien den Kennsatz bereitstellen muss. Für Ausgabedateien kann OPENV zum Positionieren oder Schreiben von (maximal 9) Benutzerkennsätzen (UVL) verwendet werden. Der Benutzer kann alle erforderlichen BTAM-Makroaufrufe absetzen.

Für Eingabedateien mit Nichtstandardkennsätzen:
Der Benutzer kann am OPENV-Ausgang seine Bandanfangskennsätze (wenn vorhanden) einlesen und prüfen.

Bevor der Benutzer die Kontrolle an das System zurückgibt (EXRTN-Makroaufruf), muss er das Band vor die Anfangskennsätze (HDR) der Datei – falls vorhanden – positionieren. Das Positionieren kann auch über FSEQ (siehe FILE/FCB) erfolgen. Der Benutzer ist für die korrekte Positionierung verantwortlich. Während des Positionierens kann das Benutzerprogramm keinen Bandwechsel durchführen.

OPENX
Der FCB wurde im Laufe der OPEN-Verarbeitung bereits auf Grund von Angaben in der TFT oder dem Katalog aktualisiert. Das Programm kann nun sicherstellen, dass alle Parameter so vorliegen, dass der OPEN ohne Fehler beendet werden kann. Bei OPEN= OUTPUT/OUTIN ist der FCB zwar über die TFT aktualisiert, der Katalogeintrag jedoch noch nicht. Die Verarbeitung wird nach Aufruf des EXRTN-Makros fortgesetzt.

OPENZ
Bei Dateien, die OUTPUT oder OUTIN eröffnet werden, ist zu dem Zeitpunkt, an dem der OPENZ-Ausgang genommen wird, die Katalogverarbeitung bereits abgeschlossen; es muss noch die abschließende OPEN-Verarbeitung durchgeführt werden.

In der OPENZ-Routine kann der FCB so modifiziert werden, dass die Verarbeitung durchgeführt werden kann: z.B. kann der Anwender eine Datei mit einer anderen als der im Katalog eingetragenen Zugriffsmethode bearbeiten.

Nach Aufruf des EXRTN-Makros wird die Verarbeitung fortgesetzt.

PASSER
Für eine geschützte Datei wurde ein falsches Kennwort angegeben.

PGLOCK
Nur zusammen mit SHARUPD=YES für UPAM oder ISAM:
vom aufrufenden Auftrag angeforderte Sperren können nicht gesetzt werden, weil sie bereits von einem anderen Auftrag gesetzt wurden; es besteht jedoch keine Deadlock-Gefahr.

UPAM:
die im FCB-Operanden PAMTOUT bestimmte Wartezeit ist abgelaufen, wenn dieser Ausgang angesprungen wird.

ISAM:
ist eine PGLOCK-Routine vorhanden, wartet ISAM nicht auf eine Satzsperre. Gilt PGLOCK=NO wird der Auftrag in eine Warteschlange eingereiht, der Benutzer wird dann nicht davon informiert, dass der Satz bereits von einem anderen Auftrag gesperrt wurde.

Wurde eine Datei mit SHARUPD=YES eröffnet, kann bei allen ISAM-Makroaufrufen die Steuerung an diesen Ausgang übergeben werden (ausgenommen OSTAT). Wenn der Ausgang PGLOCK genommen wird, ist der interne Zeiger falsch, es sei denn, der Zustand wurde durch PUTX- oder ELIM-Makroaufruf (ohne KEY) verursacht. Daher ist es unbedingt erforderlich, diesen Zeiger zurückzupositionieren, bevor ein Makroaufruf ausgegeben wird, der voraussetzt, dass der Zeiger richtig steht (z.B. bei GET, GETR und GETFL). Der Zeiger kann zurückpositioniert werden mit dem RETRY-Makroaufruf oder neu positioniert werden mit einem der ISAM-Aktionsmakroaufrufe GETKY, SETL, PUT, STORE, INSRT oder ELIM (mit KEY). Werden GET, GETR oder GETFL aufgerufen, bevor der Zeiger zurückpositioniert wurde, wird die Steuerung an den USERERR-Ausgang übergeben.

Wurde der Zustand, der dazu führte, dass der PGLOCK-Ausgang genommen wurde, durch PUTX- oder ELIM-Makroaufruf (ohne KEY) verursacht, bleibt der Datenblock gesperrt und ein Zurückpositionieren ist nicht erforderlich.

SEQCHK
Für ISAM:
ein Satz, der mit einem PUT-Makroaufruf in eine ISAM-Datei aufgenommen werden soll, enthält einen Satzschlüssel, der kleiner ist als der höchste Schlüssel der vorhandenen ISAM-Datei.

USERERR
Das Programm versucht eine unzulässige oder fehlerhafte Aktion auszuführen, wie z.B. Schreiben in eine mit INPUT eröffnete Datei, Aufruf eines PUTX- oder ELIM-Makros (ohne Schlüssel) für eine SHARUPD eröffnete ISAM-Datei ohne vorherige Sperre, unzulässiger PAM-Operationsschlüssel u.a.

WLRERR
Ein Satz mit falscher Länge wurde gelesen. Bei geblockten Sätzen fester Länge wird die Satzlänge als falsch betrachtet, wenn die Blocklänge nicht ein Vielfaches der im FCB-Eintrag RECSIZE definierten Satzlänge ist, bis zu der im FCB-Eintrag BLKSIZE definierten maximalen Blocklänge. Somit ist es zulässig, kurze Blöcke logischer Sätze zu lesen, ohne die Anzeige „falsche Satzlänge“ zu erhalten. Bei Sätzen variabler Länge ist die Satzlänge falsch, wenn sie sich nicht mehr mit der angegebenen Satzlänge im Steuerfeld des Blockzählers deckt.

Für Dateien mit Satzformat U braucht kein WLRERR-Ausgang vorgesehen zu werden, da die Satzlänge bei diesen Sätzen nicht überprüft wird.

Wird einem Programm an diesem Ausgang die Steuerung übertragen, enthält das Mehrzweckregister 0 die Adresse des fehlerhaften Blocks. Soll der Verarbeitungsablauf fortgesetzt werden, so muss der EXRTN-Makro aufgerufen werden. Enthält das Mehrzweckregister 0 den Wert X'00000001', wird der derzeitige Block übersprungen und die Verarbeitung ab dem nächsten Block fortgesetzt. Jede andere Anzeige bedeutet, dass der Block so verarbeitet werden soll, als ob kein Fehler aufgetreten wäre.

Ist für WLRERR keine separate Fehlerbehandlung vorgesehen, prüft das DVS beim Eintreten des Ereignisses „falsche Satzlänge“ die Angaben im ERROPT-Operanden:

– ERROPT ≠ NO: der Satz mit „falscher Länge“ wird behandelt wie ein fehlerhafter Block und die Steuerung an die bei ERROPT genannte Adresse übergeben.
– ERROPT = NO: der Auftrag wird abgebrochen.

Hinweise zur Programmierung

1. Die Register 14, 15, 0 und 1 sind DVS-Parameter-Register. Es gilt deshalb: falls nicht explizit anders beschrieben, kann der Anwender nicht davon ausgehen, dass diese Register einen definierten Wert haben, wenn er beim EXLST-Ausgang die Kontrolle erhält.
2. Die folgenden Tabellen zeigen, wann welche EXLST-Ausgänge verwendet werden:

Fehler-ausgang	STD SAM	NSTD SAM	ISAM	PAM	BTAM	Der Anwender kann angeben		
						CLOSE	EXRTN	Aktions-Makroaufruf
CLOSER	A	A	A	A	A	N	N	N
CLOSPOS	N	A	N	A	A	N	A	A
DLOCK	N	N	N	A	N	A	N	A
DUPEKY	N	N	A	N	N	A	N	A
EOFADDR	A	A	A	A(1)	A(2)	A	N	A
EOVCTRL	A	A	N	N	A	A	A	N
ERRADDR	A	A	A	A	A	A	N	A
ERROPT	A	A	N	N	N	A	A	N
ISPERR	N	N	A	N	N	A	N	A
LABEND	A	A	N	A	A	X	A	A(3)
LABEOV	A	A	N	A	A	X	A	A(3)
LABERR	A	X	N	A	A	X	A	A(3)
LABGEN	A	A	N	A	A	X	A	A(3)
LOCK	A	A	A	A	N	N	N	A
NODEV	A	A	A	A	A	N	N	N
NOFIND	N	N	A	N	N	A	N	A
NOSPACE	A	N	A	N	N	A	N	A/N(SAM)
OPENC	A	N	A	A	N	N	N	sinnvoll VERIF
OPENER	A	A	A	A	A	N	N	N
OPENV	A	A	N	A	A	X	A	A(3)
OPENX	A	A	A	A	A	A	A	N
OPENZ	A	A	A	A	A	A	A	N
PASSER	A	A	A	A	A	N	N	N
PGLOCK	N	N	A	A	N	A	N	A
SEQCHK	N	N	A	N	N	A	N	A

Fehler-ausgang	STD SAM	NSTD SAM	ISAM	PAM	BTAM	Der Anwender kann angeben		
						CLOSE	EXRTN	Aktions-Makroaufruf
USERERR	A	A	A	A	A	A	N	A
WLRERR	A	A	N	N	N	A	A	N

Bedeutung der Einträge

EXLST-Operand	COMMON-Ausgang	FCB-Schlüssel (Feld ID1XITB) X' '	1) Programm-abbruch	Bedeutung
CLOSER	Z	2C	A	Datei wird als geschlossen angesehen
DLOCK	Z	3C	A	Keine weiteren Sperrungen bis alle Sperrungen aufgehoben sind
DUPEKY	Z	54	A	
EOFADDR	Z	40	A	
EOVCTRL	N	34	N	
ERRADDR	Z	44	A	Fehlerschlüssel und/oder Endebyte (PAM und BTAM)
ERROPT	N	48	A	Endebyte gespeichert im FCB
ISPERR	Z	50	A	
LABEND	N	30	N	Datei wird als geschlossen angesehen
LABEOV	N	28	N	-
LABERR	N	6C	N	Nur für Dateien mit Standardkennsätzen
LABGN	N	24	N	-
LOCK	Z	10	A	
NODEV	Z	14	A	
NOFIND	Z	58	A	
NOSPACE	Z	4C	A	
OPENC	N	68	N	
OPENER	Z	08	A	Fehlercode im FCB gespeichert
OPENV	N	1C	N	
OPENX	N	04	N	
OPENZ	N	18	N	
PASSER	Z	0C	A	
PGLOCK	Z	38	A	

EXLST-Operand	COMMON-Ausgang	FCB-Schlüssel (Feld ID1XITB) X' '	1) Programm-abbruch	Bedeutung
SEQCHK	Z	60	A	
USERERR	Z	5C	A	Fehlerschlüssel gespeichert im FCB
WLRERR	N	64	N	Endebyte im FCB gespeichert

Legende

A zulässig

X nicht zulässig

Z zutreffend

N nicht zutreffend

(1) nur bei der *DUMMY-Datei

(2) nur bei FEOV verwendet

(3) nur bei Angabe von LABEL=NSTD

1) Das Programm wird beendet, falls der Ausgang nicht angegeben wurde und das Ereignis eintritt

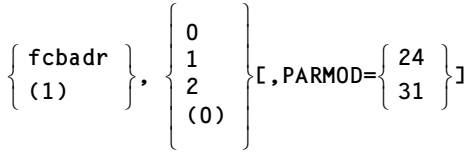
EXRTN – Rücksprung aus Fehlerroutinen

Makrotyp: R-Typ

Der EXRTN-Makroaufruf wird in einigen Anwenderroutinen benötigt, die über EXLST-Ausgänge adressiert werden. Er gibt die Steuerung an das DVS zurück, das den Funktionsschlüssel auswertet und die Verarbeitung entsprechend fortsetzt.

Der EXRTN-Makroaufruf muss angegeben werden in Routinen zu folgenden EXLST-Ausgängen: CLOSPOS, EOVCTRL, ERROPT, LABEND, LABEOV, LABERR, LABGN, OPENV, OPENX, OPENZ, WRLERR. Bei Behandlung von UHL-, UTL- und UVL-Kennsätzen muss der Makroaufruf LBRET verwendet werden.

Format

Operation	Operanden
EXRTN	

Operandenbeschreibung

fcbadr
symbolische Adresse des FCBs der Datei, deren Verarbeitung zu einer Programmverzweigung über einen EXLST-Ausgang führte

(1)
Register 1 enthält die FCB-Adresse.

Der zweite Operand gibt einen Funktionsschlüssel an; er ist nur bei den EXLST-Ausgängen ERROPT und WRLERR von Bedeutung.

0
Der Fehler wird ignoriert.

1
Plattendateien: der aktuelle Block ist zu übergehen und der Nächste zu verarbeiten.
Banddateien: Das Benutzerprogramm soll mit einem OPEN- bzw. Bandendefehler beendet werden.

2
Banddateien: Die Bandende-Verarbeitung soll fortgeführt werden (nur beim LABERR-Ausgang).

(0)
Register 0 enthält im rechten Byte den Funktionscode.

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.

Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst.
Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

Hinweis zur Programmierung

Der EXRTN-Makroaufruf zerstört die Register 0, 1, 14 und 15.

FCB – Dateisteuerblock definieren

Makrotyp: O-Typ

Der Dateisteuerblock ist die zentrale Informationsquelle für die Zugriffsmethoden BTAM, ISAM, SAM und UPAM. Bei jeder Bearbeitung einer Datei wird ein Dateisteuerblock (FCB) benötigt.

Die nötigen Informationen entnimmt das DVS verschiedenen Quellen:

– Im FCB-Makroaufruf kann der Anwender bereits FCB-Felder versorgen.

– Im Anwenderprogramm können vor der Dateieröffnung FCB-Felder während des Programmlaufs versorgt werden.

– Angaben im FILE-Makroaufruf mit entsprechendem Linknamen werden in die TFT übernommen und haben bei der Dateieröffnung Vorrang vor den entsprechenden FCB-Feldern.

– Während der Dateieröffnung kann der FCB auch durch Routinen des Anwendungsprogramms verändert werden (siehe Makro EXLST, Seite 153, OPENX- und OPENZ-Routinen).

– Weitere Informationen werden bei der OPEN-Verarbeitung dem Katalogeintrag der Datei entnommen.

In chronologischer Reihenfolge werden die Ereignisse aufgelistet, die zum Aufbau eines kompletten FCB führen:

– FCB-Makroaufruf während der Assemblierung

– Modifizierung während des Programmablaufs vor dem OPEN-Makro

– zum OPEN-Zeitpunkt Aufbau des FCB

  – aus der TFT (FILE-Aufruf mit entsprechendem Linknamen)
  – aus dem Katalogeintrag (unabhängig vom OPEN-Modus)
  – in OPEN-Routinen (siehe Makro EXLST: OPENX- und OPENZ-Routinen) durch das Anwenderprogramm

Der komplette FCB stellt schließlich auch die Verbindung zu den logischen Routinen her, die bei den satzorientierten Zugriffsmethoden (SAM, ISAM) das Blocken und Entblocken der Datensätze übernehmen.

DSECTs

Mit dem Makro IDFCB kann eine DSECT für den Dateisteuerblock generiert werden, sodassass der Anwender die FCB-Felder symbolisch adressieren kann. Werden Dateien über die 24-Bit-Schnittstelle verarbeitet, kann mit dem Makroaufruf IDFCBE eine DSECT für die FCB-Erweiterung generiert werden.

NULL-Operanden

Existiert die im FCB-Makroaufruf direkt oder über den LINK-Operanden indirekt spezifizierte Datei/Dateigenerationsgruppe bereits (Datei wurde schon einmal mit OPEN-Mode OUTPUT oder OUTIN eröffnet), können einige Operanden des FCB-Makros als so genannte „NULL-Operanden“ angegeben werden. Das bedeutet, dass zwar der Operand, aber kein Operandenwert angegeben wird (leere Zeichenfolge als Operandenwert).

FCB LINK=name, FCBTYPE=, BLKSIZE=, RECFORM=, ...

Welche Operanden als NULL-Operanden angegeben werden können, kann der Tabelle „Operanden für Plattendateien“ auf Seite 447 entnommen werden.

FCB und Zugriffsmethoden

Verschiedene Zugriffsmethoden benutzen den FCB. Es werden allerdings jeweils nur bestimmte Operanden ausgewertet. Welche Zugriffsmethode welche Operanden auswertet, wird in der nachfolgenden Übersichtstabelle dargestellt. Operanden, die die Zugriffsmethode nicht auswerten kann, werden ignoriert. Es wird keine Fehlermeldung ausgegeben.

Operand im FCB	**Bedeutung**	**Zugriffsmethode**			
		BTAM	**ISAM**	**PAM**	**SAM**
BLIM	nur Band: Anzahl der Datenblöcke pro Band				x
BLKCTRL	Blockformat (UPAM: Band)		x	x	x
BLKSIZE	Größe des Datenblocks (UPAM: Band)	x	x	x	x
BTAMRQS	Anzahl der Ein-/Ausgabe-Aufträge	x			
BUFOFF	nur Band: Pufferverschiebung				x
CHAINIO	Kettungsfaktor	x			
CHKPT	nur Band: Fixpunkt				x
CODE	Umsetzungstabelle (SAM: Band)	x			x
DUPEKY	Mehrfachschlüssel		x		
EXIT	Fehlerausgang	x	x	x	x
FCBTYPE	Zugriffsmethode	x	x	x	x
FILE	bezeichnet die zu verarbeitende Datei	x	x	x	x
FORM	Speicherplatzreservierung				x

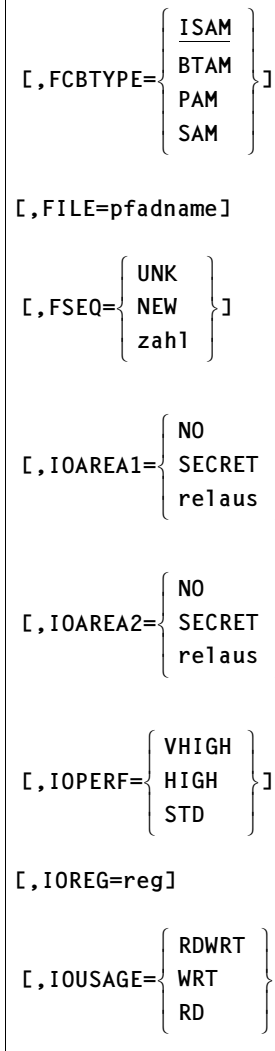
Operand im FCB	Bedeutung	Zugriffsmethode			
		BTAM	**ISAM**	**PAM**	**SAM**
FSEQ	nur Band: Nummer einer Datei innerhalb einer Dateimenge	x			x
IOAREA1	Programmpuffer	x	x	x	x
IOAREA2	zweiter Programmpuffer	x	x	x	x
IOPERF	Performanceattribut (nur Pubset)		x	x	x
IOREG	Register für Dateiverarbeitung im Locate-Mode		x		x
IOUSAGE	Nutzung des Cache (nur Pubset)		x	x	x
KEYARG	Adresse des Feldes, das den ISAM-Schlüssel enthält		x		
KEYLEN	Länge des ISAM-Schlüssel		x		
KEYPOS	Beginn des ISAM-Schlüssels		x		
LABEL	nur Band: Kennsatzeigenschaften	x			x
LARGE_ FILE	nur Platten: Dateigrößenallokierung über 32 GB		x	x	x
LINK	Dateikettungsnamen	x	x	x	x
LOCKENV	Lockprotokoll zur Synchronisation (Shared-Update-Verarbeitung)		x	x	
LOGLEN	Länge der logischen Markierung		x		
OPEN	OPEN-Modus	x	x	x	x
OPTION	Optionen	x	x	x	x
OVERLAP	Überlappung		x		
PAD	Blockfüllung bei sequenzieller Dateierstellung		x		
PAMREQS	asynchrone Ein-/Ausgaben			x	
PAMTOUT	Wartezeit			x	
PARMOD	Generierungsmodus	x	x	x	x
PASS	Kennwort	x	x	x	x
POOLLNK	Poolkettungsname		x		
RECFORM	Satzformat	x	x		x
RECSIZE	Satzlänge	x	x		x
RETPD	Schutzfrist	x	x	x	x
SECLEV	nur Band: Security Level	x			x
SHARUPD	Multi-User-Betrieb		x	x	
STREAM	Streaming-Modus	x			
TAPEWR	nur Band: gepufferte Ausgabe	x			x

Operand im FCB	Bedeutung	Zugriffsmethode			
		BTAM	ISAM	PAM	SAM
TPMARK	nur Band: Abschnittsmarken	x			x
TRANS	nur Band: Code-Umsetzung	x			x
TRTADR	nur Band: benutzereigene Übersetzungstabelle (Lesen)	x			x
TRTADW	nur Band: benutzereigene Übersetzungstabelle (Schreiben)	x			x
VALLEN	Länge der Wertmarkierung		x		
VALPROP	Wertmarkierung		x		
VARBLD	Register für den freien Platz im schreibenden Block				x
WRCHK	Kontroll-Lesen beim Schreiben von Blöcken		x	x	x
WROUT	sofortiges Zurückschreiben		x		

Format

Operation	Operanden
FCB	[BLIM=zahl] [,BLKCTRL={PAMKEY | DATA | DATA2K | DATA4K | NO}] [,BLKSIZE={STD | (STD,n) | länge}] [,BTAMRQS=zahl] [,BUFOFF={L | länge}] [,CHAINIO=zahl] [,CHKPT={NO | ANY | BLIM | FEOV}, {ACTIVE | DUMMY}] [,CODE={EBCDIC | ISO7 | OWN}] [,DUPEKY=YES] [,EXIT={(relaus) | relaus}]

(Teil 1 von 5)

Operation	Operanden
	

(Teil 2 von 5)

Operation	Operanden
	[,LABEL={(STD,3) / STD / (STD,zahl) / NO / NSTD}] [,LARGE_FILE={*FORBIDDEN / *ALLOWED}] [,LINK=name] [,LOCKENV={*HOST / *XCS}] [,LOGLEN=länge] [,OPEN={INPUT / EXTEND / INOUT / OUTIN / OUTPUT / REVERSE / SINOUT / UPDATE}] [,OPTION={code / (code1,code2)}] [,OVERLAP=YES] [,PAD=zahl] [,PAMREQS=zahl] [,PAMTOUT=zahl] [,PARMOD={24 / 31}] [,PASS=kennwort] [,POOLLNK=name] [,RETPD=tage]

(Teil 3 von 5)

Operation	Operanden
	[,SHARUPD={NO / YES / WEAK}]
	[,RECFORM={{V / F / U} / {V / F / U}[,{N / M / A}])}]
	[,RECSIZE={länge / reg}]
	[,SECLEV={HIGH / LOW / ({HIGH / LOW},OPR)}]
	[,STREAM={NO / YES}]
	[,TAPEWR={DEVICE-BUFFER / IMMEDIATE}]
	[,TPMARK={YES / NO}]
	[,TRANS={YES / NO}]
	[,TRTADR=relaus]
	[,TRTADW=relaus]
	[,VALLEN=länge]
	[,VALPROP={MIN / MAX}]

(Teil 4 von 5)

Operation	Operanden
	[,VARBLD=reg] [,WRCHK={NO / YES}] [,WROUT={NO / YES}]

(Teil 5 von 5)

Operandenbeschreibung

BLIM = zahl

Nur zum Erstellen von Banddateien mit Standardkennsätzen, die mit der Zugriffsmethode SAM verarbeitet werden sollen und sich über mehrere Bänder erstrecken:
Gleichzeitig mit dem Operanden BLIM müssen folgende Operanden im FCB-Makroaufruf angegeben werden: FCBTYPE=SAM, OPEN=OUTPUT, LABEL=(STD,n).
„zahl“ gibt an, wie viele Datenblöcke auf ein Band geschrieben werden dürfen,
$1 \leq$ zahl ≤ 999999.
Bei Erreichen des Grenzwerts wird Bandwechsel veranlasst (EOV-Verarbeitung); falls mit CHKPT-Operand gefordert, wird zuvor noch ein Fixpunkt an das Bandende geschrieben. Ist das Bandende erreicht, bevor die mit BLIM festgelegte Anzahl Blöcke geschrieben wurde, wird ein Fehlercode im FCB eingetragen.

BLKCTRL

Wirksam nur bei 31-Bit-Verarbeitung (PARMOD=31); Operand wird bei 24-Bit-Verarbeitung (PARMOD=24) ignoriert:
Legt fest, ob eine Datei des K-Formats (mit PAM-Schlüssel) oder des NK-Formats (ohne PAM-Schlüssel) zu verarbeiten ist.
Für die Verarbeitung von NK-SAM- und NK-PAM-Dateien stehen die gleichen Funktionen mit identischen Anwenderschnittstellen zur Verfügung wie für die entsprechenden K-Dateien. Die Zugriffsmethode NK-ISAM bietet einige Funktionen an, die über die des K-ISAM hinausgehen (zu ISAM-Pool und Sekundärschlüssel siehe Handbuch „Einführung in das DVS“ [1]).

Um eine existente Datei zu eröffnen, sollte für BLKCTRL kein Wert angegeben werden (NULL-Operand). (Erläuterungen zu dem Begriff NULL-Operand, siehe Seite 411). Der Wert wird in diesem Fall zum OPEN-Zeitpunkt aus dem Katalogeintrag in den FCB übernommen (zu „existenten“ Dateien siehe auch den Hinweis bei Abschnitt „Ablauf der OPEN-Verarbeitung“ im Kapitel „OPEN-Verarbeitung“ im Handbuch „Einführung in das DVS“ [1]).
Bei Neuanlegen einer Datei (OPEN-Modus-OUTPUT oder OUTIN), sollte der Operand

BLKCTRL nicht angegeben werden. Während der OPEN-Verarbeitung wird dann in Abhängigkeit der Dateistruktur und des Plattenformats folgende **Voreinstellung** für BLKCTRL vorgenommen:

BLKCTRL	**Dateistruktur – Plattenformat**
PAMKEY	für Dateien (PAM, SAM, ISAM) auf K-Platten und Banddateien, falls nicht FCBTYPE=BTAM.
DATA	für SAM-Dateien auf NK2- und NK4-Platten
DATA2K	für ISAM-Dateien auf NK2-Platten
DATA4K	für ISAM-Dateien auf NK4-Platten
NO	für PAM-Dateien auf NK2- und NK4-Platten sowie für BTAM-Dateien

= PAMKEY
Die Datei hat das K-Format: Die Blockkontrollinformation wird außerhalb des Datenblocks in einem PAM-Schlüssel abgelegt. Eine solche K-Datei kann nicht auf einer NK-Platte (FBA-Platte ohne PAM-Schlüssel-Simulation) angelegt werden.

= DATA
Die Datei hat NK-Format: Die Blockkontrollinformation steht zu Beginn eines jeden logischen Blockes (siehe auch Operandenbeschreibung BLKSIZE; bei ISAM-Dateien am Beginn eines jeden 2-KByte- bzw. 4-KByte-Blockes). Eine NK-Datei kann sowohl auf K-Platte, auf NK2-Platte und bei entsprechend gewählter Blocklänge auch auf NK4-Platte liegen. Beim Neuanlegen einer Datei (OPEN OUTPUT/OUTIN) wird eine NK2- oder eine NK4-Datei erstellt:
Für Dateien, die mit anderen Zugriffsmethoden als ISAM erstellt wurden, gilt in Abhängigkeit des Blockungsfaktors „n“ bei der Angabe der logischen Blocklänge mit BLKSIZE Folgendes:
– Ist der Blockungsfaktor n eine ungerade Zahl wird eine NK2-Datei angelegt
– Ist der Blockungsfaktor n eine gerade Zahl wird eine NK4-Datei angelegt

Beim Erstellen einer NK-ISAM-Datei (OPEN OUTPUT/OUTIN) wird das Dateiformat in Abhängigkeit des Plattenformats gewählt. Eine bereits geöffnete Datei kann unabhängig vom Blockformat geöffnet werden.

= DATA2K
Nur für ISAM-Dateien:
Es wird explizit eine NK2-Datei erstellt. bzw. eine NK2-Datei bearbeitet. Die Datei kann nicht auf NK4-Platte angelegt werden. Eine Datei, die auf NK4-Platte liegt, kann mit dieser Angabe nicht eröffnet werden.
Die blockspezifische Verwaltungsinformation wird in den ersten 16 Byte eines jeden 2-KByte-Blocks hinterlegt.

= DATA4K

Nur für ISAM-Dateien:
Es wird explizit eine NK4-Datei erstellt bzw. eine NK4-Datei bearbeitet. Die blockspezifische Verwaltungsinformation wird in den ersten 16 Byte eines jeden 4-KByte-Blocks hinterlegt. Wird ein Blockungsfaktor n angegeben, so muss n eine gerade Zahl sein. d.h. die logische Blockgröße muss ein Vielfaches von 4-KByte betragen (BLKSIZE=(STD,n) mit n gerade Zahl).
Die Datei kann auf K- NK2- und NK4-Platte angelegt werden bzw. dort eröffnet werden.

= NO

Die Angabe ist nur für PAM-Dateien und SAM-Banddateien sinnvoll; bei SAM-Plattendateien wird sie in BLKCTRL=DATA, bei ISAM-Dateien in BLKCTRL=DATA2K bzw. BLKCTRL=DATA4K umgewandelt.
Wird FCBTYPE=PAM angegeben, so wird eine NK-PAM-Datei angelegt, die keine blockspezifischen Verwaltungsinformationen enthält.
Diese Datei kann unabhängig von der gewählten logischen Blocklänge (BLKSIZE) sowohl auf K-Platte als auch auf NK2-Platte angelegt werden.
Wird als logische Blocklänge (BLKSIZE) ein Vielfaches von 4K angegeben (Blockungsfaktor „n“ geradzahlig), kann die Datei auch auf einer NK4-Platte angelegt werden.

BLKSIZE

legt die Länge des logischen Blocks (Datenblocks) fest, d.h. die Länge der Übertragungseinheit von und zu den Ein-/Ausgabegeräten und damit die Länge des Ein-/Ausgabebereichs des Programms.

Wird für BLKSIZE **keine** Angabe gemacht, so wird für existente Dateien (Definition siehe auch den Hinweis bei Abschnitt „Ablauf der OPEN-Verarbeitung“ im Kapitel „OPEN-Verarbeitung“ im Handbuch „Einführung in das DVS“ [1]) der Wert aus dem Katalogeintrag übernommen. Bei Neuanlegen werden (STD,2)-Dateien auf NK4-Datenträgern angelegt. Auf anderen Datenträgern werden (STD,1)-Dateien angelegt.
Der Benutzer darf, falls (STD,2) eingestellt wird, keine eigenen IOAREAs verwenden. Diese sollte er zum OPEN-Zeitpunkt vom System im Klasse-5-Speicher anlegen lassen.

Für die Verarbeitung siehe Hinweis unter „BLKCTRL“ auf Seite 418 und bei „Hinweise zur Programmierung“ auf Seite 446.

Plattendateien/Banddateien mit Standardblöcken:
Ein logischer Block kann aus mehreren PAM-Seiten bestehen. Das System verknüpft die zu einer Übertragungseinheit zusammengefassten PAM-Seiten automatisch.

Für Plattendateien ergeben sich Wechselwirkungen mit dem RECSIZE-Operanden, für Banddateien mit dem LABEL-Operanden.

Banddateien mit Nichtstandardblöcken:
Der Datenblock ist definiert durch die Anzahl Bytes, die pro Schreib- bzw. Leseoperation geschrieben/gelesen werden.

= STD
entspricht der Angabe (STD,1); siehe unten. Die Daten werden in Einheiten von 2048 Byte von/zu den Geräten übertragen; die für Anwenderdaten nutzbare Länge der Übertragungseinheit ist abhängig von der BLKCTRL-Angabe (bzw. dem Plattentyp).

= (STD,n)
„STD“ ist ein Standardblock der Größe 2048 Byte; „n“ ist der Blockungsfaktor ($1 \leq n \leq 16$).

Jeder logische Block besteht aus n PAM-Blöcken (1 PAM-Block/1 PAM-Seite = 2048 Byte), d.h. die maximale Länge des logischen Blocks ist 16 PAM-Seiten = 32768 Byte. Für NK-Dateien:
„n“ legt die Länge des logischen Blockes als Vielfaches von 2048 Byte fest: Die Länge eines solchen Blockes beträgt n * 2048 Byte.
Für NK4-Dateien muss „n“ eine gerade Zahl sein, d.h. die Länge des logischen Blocks ist ein Vielfaches von 4-KByte. Für eine NK4-ISAM-Datei muss zusätzlich für den Operanden BLKCTRL der Operandenwert DATA4K angegeben sein.

Für SAM-Dateien, SETL-Verarbeitung: in jedem logischen Block dürfen höchstens 255 Sätze stehen, da die Positionierungsinformation nur 1 Byte lang ist. Diese Einschränkung entfällt bei der Verwendung eines 31-Bit-fähigen FCB.

BLKCTRL	**für Anwenderdaten nutzbare Blocklänge (in Byte)**	
= PAMKEY		n * 2048
= DATA	bei ISAM: bei SAM: bei PAM:	n * (2048 - 16) - 16 (n * 2048) - 16 (n * 2048) - 12
= DATA2K / DATA4K	nur für ISAM möglich:	n * (2048 - 16) - 16}
= NO		n * 2048

Hinweis

Bei der Frage, ob ein Satz in einen Block passt oder wie viele Sätze in einen Block passen, sind zusätzlich folgende Punkte zu berücksichtigen:
- Für NK-ISAM-Dateien mit Mehrfachschlüsseln die Länge des Zeitstempels
- Für NK-ISAM-Dateien mit RECFORM=F die Länge des Satzformatfeldes
- Für NK-SAM-Dateien die Länge des Längenfeldes (Füllgradinformation)

= länge

Nur für Banddateien:

gibt die maximale Blocklänge in Byte an und legt gleichzeitig fest, dass die Datei aus Nichtstandardblöcken besteht, d.h. es werden keine PAM-Schlüssel geführt. Es sind zum einen die Operanden BUFOFF und RECFORM zu berücksichtigen, zum anderen FCBTYPE und CHAINIO.

RECFORM	**Auswirkung**
RECFORM=F	„länge“ gibt die Blocklänge einschließlich Länge der Pufferverschiebung an (siehe Operand (BUFOFF); alle Blöcke haben dieselbe Länge
RECFORM=V/U	„länge“ gibt die maximale Blocklänge einschließlich der Länge der Pufferverschiebung (siehe Operand BUFOFF) an, d.h. die Blocklänge ist (wie die Satzlänge) variabel gilt RECFORM=V zusammen mit CODE=EBCDIC oder LABEL=(STD,n) mit n>1, muss „länge“ < 10000 sein (interne Umwandlung in Satzformat D)

FCBTYPE	**zulässige Angabe für „länge“**
SAM, BTAM	$1 \leq n \leq 32768$
PAM	--------

Hinweis

Soll eine bestehende Datei eröffnet werden, so wird empfohlen, den Nulloperanden zu verwenden. Der Wert wird aus dem Katalogeintrag zum OPEN-Zeitpunkt übernommen.
Soll eine Datei neuangelegt werden, so muss für NK4-Datenträger die Angabe BLKSIZE=(STD,n) mit $1 \leq n \leq 16$ und n gerade erfolgen, ansonsten wird der OPEN abgewiesen.

BTAMRQS = zahl

Nur für BTAM:

gibt die Anzahl der Ein-/Ausgabe-Aufträge für BTAM an, die direkt nacheinander (ohne WAITs) an das System abgegeben werden können (MAV-Modus). Alle angenommenen Aufträge befinden sich gleichzeitig zur Bearbeitung im System. Es ist gewährleistet, dass eine serielle Bearbeitung erfolgt. Die Verarbeitung erfolgt also asynchron.

$1 \leq$ zahl ≤ 8.

Voreinstellung: BTAMRQS = 1

Die Zugriffsmethode BTAM ist im Handbuch „Einführung in das DVS“ [1] beschrieben.

BUFOFF
Nur für SAM-Banddateien ohne Standardblockung oder Banddateien mit BLKCTRL=DATA:
legt die Pufferverschiebung (Buffer Offset) fest, d.h. die Länge eines Feldes, das am Anfang eines jeden Datenblocks eingefügt wird.

Voreinstellung:
Wird der Operand BUFOFF nicht angegeben (weder TFT noch FCB), so wird der Datei nach dem Eröffnen folgender Wert zugewiesen (sofern nicht der Wert aus dem Katalogeintrag übernommen wird):

– für Banddateien mit BLKCTRL=DATA

  bei FCBTYPE=SAM: BUFOFF=16
  bei FCBTYPE=PAM: BUFOFF=12

– für SAM-Banddateien ohne Standardblockung

  bei RECFORM=V: BUFOFF=4
  bei RECFORM=F/U: BUFOFF=0

  = L
  Der BUFOFF-Wert wird dem HDR2-Kennsatz der Datei entnommen. Ist kein HDR2-Kennsatz vorhanden oder enthält das Feld „Pufferverschiebung“ Leerzeichen (X'4040'), tritt die Voreinstellung in Kraft.

  = länge
  gibt die Länge der „Pufferverschiebung“ an.

  Für Dateien mit RECFORM=V gilt: 0 ≤ länge ≤ 4; ist BUFOFF=4, enthält dieses Feld die aktuelle Blocklänge.

CHAINIO = zahl
Nur für BTAM-Dateien bei geketteter Ein-/Ausgabe:
1 ≤ zahl ≤ 16; „zahl“ ist der Kettungsfaktor, der die Länge der Transporteinheit bei der Ein-/Ausgabe festlegt. „zahl“ bezeichnet dabei eine Anzahl Blöcke, sodass sich die Länge der Transporteinheit berechnet aus „zahl“ * Blocklänge.

Eine Angabe im Operanden LEN im BTAM-Aktionsmakroaufruf hat Vorrang gegenüber dem Produkt Blockgröße * zahl; dennoch muss CHAINIO angegeben werden, wenn mit „Kettung“ gearbeitet wird.

CHKPT
Für Banddateien:
steuert, ob und wann automatisch ein Fixpunkt an das Bandende zu schreiben ist oder wie die Datei bei Wiederanlauf weiterverarbeitet werden soll (zum Kommando RESTART-PROGRAM siehe Handbuch „Kommandos“ [3]).

Voreinstellung: CHKPT=(NO,ACTIVE)

= (NO,...)
Es erfolgt keine automatische Fixpunktschreibung.

= (BLIM,...)
Wenn das mit dem BLIM-Operanden gesetzte Blocklimit erreicht ist, wird automatisch ein Fixpunkt geschrieben; der Operand BLIM muss angegeben werden.

= (FEOV,...)
Bei jedem FEOV-Makroaufruf wird automatisch ein Fixpunkt geschrieben.

= (ANY,...)
Ein Fixpunkt wird automatisch geschrieben, wenn die mit BLIM gesetzte Grenze erreicht ist oder ein FEOV-Makro aufgerufen wird; der Operand BLIM muss angegeben werden.

= (...,DUMMY)
Die Datei „pfadname“ wird bei einem Wiederanlauf mit dem Kommando RESTART-PROGRAM wie eine DUMMY-Datei behandelt.

= (...,ACTIVE)
Die Datei „pfadname“ wird bei einem Wiederanlauf (Kommando RESTART-PROGRAM) weiterverarbeitet.

CODE
Für Bandverarbeitung:
legt fest, ob und welche Umsetzungstabellen bei Ein-/Ausgabe verwendet werden.

Bei CODE=EBCDIC und CODE=ISO7 haben deutscher und internationaler Zeichensatz die gleiche Verschlüsselung.

Bei CODE=ISO7 und CODE=OWN ist Folgendes zu beachten:
– die Blocklänge muss mit BLKSIZE=länge definiert werden, damit kein PAM-Schlüssel geschrieben wird;
– bei Ausgaben im Locate-Mode ändert sich bei variablem Satzformat (RECFORM=V) der Inhalt des Satzlängenfeldes

= EBCDIC
Bei der Verarbeitung ist keine Code-Umsetzung erforderlich.

= ISO7
Die Banddatei ist/wird mit dem ISO-7-Bit-Code geschrieben, d.h. bei der Ausgabe wird EBCDI-Code in ISO-7-Bit-Code umgesetzt, bei der Eingabe ISO-7-Bit-Code in EBCDI-Code.

= OWN
Die Umsetzung erfolgt über vom Benutzer erstellte Tabellen, deren Adressen im Dateisteuerblock enthalten sein müssen. Gleichzeitig muss im LABEL-Operanden Kennsatzverarbeitung ausgeschaltet werden (LABEL=NO) oder mit LABEL=NSTD die Kennsatzverarbeitung ins Benutzerprogramm verlagert werden.

DUPEKY = YES
Für ISAM-Dateien:
haben mehrere Sätze den gleichen Primärschlüsselwert, überschreiben sie sich nicht gegenseitig, sondern werden in der Reihenfolge ihrer Erstellung hintereinander geschrieben. Der Operand DUPEKY=YES ist nur von Bedeutung, wenn die ISAM-Datei mit PUT-Makroaufruf sequenziell erstellt oder mit STORE-Makroaufruf nichtsequenziell erweitert wird. Der INSRT-Makroaufruf kann nicht dazu verwendet werden, Sätze mit gleichen Primärschlüsselwerten zu schreiben.

Voreinstellung:
Primärschlüsselwerte dürfen nicht mehrfach in der Datei vorkommen.

Bei NK-ISAM wird den Sätzen mit gleichen Primärschlüsselwerten intern ein 8-Byte-Zeitstempel angehängt, was bei der Definition der Satzlänge berücksichtigt werden sollte.

Die ISAM-Makroaufrufe PUT, STORE und INSRT wirken sich unterschiedlich aus, wenn ein Primärschlüsselwert mehrfach auftritt.

Makro-aufruf	**Mehrfachschlüssel**	
	nicht erlaubt	**erlaubt (DUPEKY=YES)**
PUT	Satz mit doppeltem Schlüssel wird nicht geschrieben; EXLST-Ausgang: DUPEKY	die Sätze werden nacheinander in die Datei aufgenommen
STORE	der „neue“ Satz überschreibt den bereits mit diesem ISAM-Schlüssel gespeicherten Satz	der neue Satz wird hinter dem alten Satz in die Datei aufgenommen
INSRT	Satz mit doppeltem Schlüssel wird nicht geschrieben; EXLST-Ausgang: DUPEKY	Satz mit doppeltem Schlüssel wird nicht geschrieben; EXLST-Ausgang: DUPEKY

Hinweis

In einer Datei mit mehrfach auftretenden Primärschlüsselwerten können keine Sekundärschlüssel definiert werden.

EXIT
Gibt die Adresse an, auf die das Programm bei einem Fehler verzweigen soll. Wird der Operand nicht angegeben, führen Ausnahmebedingungen beim Zugriff auf die Datei zur abnormalen Programmbeendigung.
Wird der Ausgang genommen, wird im FCB eine Anzeige gesetzt. Die Anwenderroutine kann feststellen, um welche Ausgangsbedingung es sich handelte. Tritt während einer Ein-/Ausgabe ein Hardwarefehler auf, wird im FCB ein 5-Byte-Feld mit Informationen des CCB versorgt (Standard-Gerätebyte, Fehlerbytes 1-3, Ablaufteil-Markierungsbyte; siehe dazu auch Makro NDWERINF, Seite 750). Im Anhang (Seite 875) sind diese Informationen näher beschrieben. Das Ablaufteil-Markierungsbyte und das Standard-Gerätebyte sind in der DSECT des CCB (IDCCB) definiert.

= (relaus)
Adresse einer Anwenderroutine im Benutzerprogramm, die die Fehlerbehandlung durchführt.

= relaus
Adresse des EXLST-Makroaufrufs, über dessen Fehlerausgänge verschiedene Anwenderroutinen adressiert werden, zur spezifischen Fehlerbehandlung.

FCBTYPE
Bestimmt die Zugriffsmethode bei der Dateiverarbeitung.

= ISAM
In Abhängigkeit vom Operanden BLKCTRL wird die Datei als NK-ISAM-Datei (BLKCTRL=DATA/DATA2K/DATA4K) oder als K-ISAM-Datei (BLKCTRL=PAMKEY) verarbeitet. Die Zugriffsmethode ISAM ist im Handbuch „Einführung in das DVS“ [1] beschrieben.

ISAM-spezifische Operanden: DUPEKY, KEYARG, KEYLEN, KEYPOS, LOGLEN, PAD, POOLLNK, VALLEN, WROUT und VALPROP.

= BTAM
Eine Banddatei wird mit der Zugriffsmethode BTAM verarbeitet (die Zugriffsmethode BTAM ist im Handbuch „Einführung in das DVS“ [1] beschrieben).

BTAM-spezifische Operanden: CHAINIO, OPEN=SINOUT, BTAMRQS

= PAM
Die Datei wird mit der Zugriffsmethode UPAM verarbeitet (siehe Beschreibung der entsprechenden Zugriffsmethode). PAM-Dateien können auf Band oder auf Platte gespeichert sein.

= SAM
Die Datei wird mit der Zugriffsmethode SAM bearbeitet. Sie kann auf Platte oder Band liegen. SAM-Dateien werden in der Regel sequenziell verarbeitet, mit den Zugriffsmethoden SAM oder auch UPAM. Die Zugriffsmethode SAM ist im Handbuch „Einführung in das DVS“ [1] beschrieben.

SAM-spezifische Operanden: BUFOFF, CLOSMSG, OPEN=UPDATE

FILE = pfadname
Bezeichnet die zu verarbeitende temporäre oder permanente Datei oder Dateigeneration mit: <c-string 1..54: filename 1..534>

Dateigenerationen können mit absoluter oder relativer Generationsnummer angesprochen werden.

Voreinstellung: FILE=fcbadr (= symbolische Adresse des FCB)
Falls nicht vorhanden: Blanks

Pfadname bedeutet [:catid:][$userid]dateiname

catid
Katalogkennung; falls nicht angegeben, wird die Default-Catid der Benutzerkennung angenommen.

userid
Benutzerkennung; falls nicht angegeben, wird die Benutzerkennung des LOGON-Kommandos angenommen.

dateiname
vollqualifizierter Dateiname

FORM = SHORT
Gilt nur für die 24-Bit-Schnittstelle (Nicht-XS-Verarbeitung):
gibt an, dass für die logischen Routinen kein Speicherplatz reserviert werden soll.

Voreinstellung: im 24-Bit-FCB wird Speicherplatz für die logischen Routinen reserviert

Die logischen Routinen übernehmen das Blocken und Entblocken der Datensätze bei SAM und ISAM; d.h. bei normaler ISAM-/SAM-Verarbeitung kann die Datei nicht eröffnet werden, wenn im FCB-Makroaufruf FORM=SHORT angegeben wurde. Für PAM-Dateien werden die logischen Routinen nicht benötigt, FORM=SHORT wird ignoriert.

Bei XS-Verarbeitung (PARMOD=31) wird FORM=SHORT ignoriert, da der 31-Bit-TU-FCB nur die Adressen der logischen Routinen enthält.

FSEQ
Für Banddateien, die zu einer Dateimenge (File Set) gehören:
gibt die (laufende) Nummer einer Datei innerhalb der Dateimenge an. Sind z.B. auf einem Band mehrere Dateien gleichen Namens gespeichert, wird der Zugriff über FSEQ gesteuert. Dies gilt auch für MF/MV-Sets.

Voreinstellung: FSEQ = 0
(Bei der Verarbeitung des Bandes wird die erste Datei bearbeitet.)

= UNK
Nur zulässig für Dateien mit Standardkennsätzen:
die Anfangsposition der Datei ist unbekannt. Das Band wird bei der Dateiverarbeitung zurückgespult.

= NEW
Nur zulässig für noch nicht katalogisierte Dateien:
die Dateimenge wird um eine neue Datei erweitert. Es wird auf das Ende der Dateimenge positioniert. Die neue Datei wird hinter die bisher letzte Datei der Dateimenge geschrieben. Die „Dateifolgenummer“ wird um 1 erhöht.

= zahl
gibt die Dateifolgenummer von „pfadname“ innerhalb einer Dateimenge an;
$0 \leq$ zahl ≤ 9999.
FSEQ=0 bezeichnet wie FSEQ=1 die erste Datei der Dateimenge.

Ist „pfadname“ bereits katalogisiert, muss die FSEQ-Angabe mit dem FSEQ-Wert im Katalogeintrag übereinstimmen. Soll eine neue Datei erstellt werden, wird sie am Dateimengenende angefügt; das heißt, die Dateifolgenummer muss um 1 höher sein als die der bisher letzten Datei der Dateimenge.

Beim Eröffnen der Datei wird das Band nicht zurückgespult (rewind), falls es schon an der mit FSEQ angegebenen Stelle steht.

IOAREA1
Gibt an, ob und an welcher Adresse ein Pufferbereich beim OPEN zugewiesen werden soll.

Voreinstellung: das DVS fordert zum OPEN-Zeitpunkt automatisch einen Pufferbereich (Klasse-5-Speicher) ober- oder unterhalb 16 MB an in Abhängigkeit von Adressierungs- und Generierungsmodus (siehe auch „Operanden IOAREA1/2“ auf Seite 449).

= NO
Zum OPEN-Zeitpunkt wird kein Pufferbereich zugewiesen (nicht zulässig bei SAM- und K-ISAM-Verarbeitung)

= SECRET
Das Datenverwaltungssystem fordert zum OPEN-Zeitpunkt für die IOAREA1 einen Bereich im nichtprivilegierten Klasse-5-Speicher an, dessen Seiten bei einer Dump-Auswertung nicht ausgegeben werden.

= relaus
Adresse eines Pufferbereichs; ist dieser Bereich kleiner oder gleich einer Seite (4096 Byte), muss er in einer Seite enthalten und auf Wortgrenze ausgerichtet sein; ist er größer, muss er auf Seitengrenze ausgerichtet sein.

IOAREA2
Gibt an, ob bei Dateieröffnung ein zweiter Pufferbereich zugewiesen werden soll.

Voreinstellung: das DVS fordert zum OPEN-Zeitpunkt automatisch einen Pufferbereich (Klasse-5-Speicher) ober- oder unterhalb 16 MB an in Abhängigkeit von Adressierungs- und Generierungsmodus (siehe auch „Operanden IOAREA1/2“ auf Seite 449).

= NO
Für die Datei wird nur ein Pufferbereich zugewiesen, überlappte Verarbeitung ist nicht möglich (vgl. Operand OVERLAP=YES); für K-ISAM-Verarbeitung kann IOAREA2=NO nicht angegeben werden.

= SECRET
Das Datenverwaltungssystem fordert zum OPEN-Zeitpunkt für die IOAREA2 einen Bereich im nichtprivilegierten Klasse-5-Speicher an, dessen Seiten bei einer Dump-Auswertung nicht ausgegeben werden.

= relaus
Adresse des zweiten Pufferbereichs. Ist er kleiner oder gleich einer Seite (4096 Byte), muss er in einer Seite enthalten und auf Wortgrenze ausgerichtet sein, ist er größer als eine Seite, muss er auf Seitengrenze ausgerichtet sein.

IOPERF
Gilt nur für die 31-Bit-Schnittstelle:
Hiermit wird das gewünschte Performanceattribut bzgl. I/O-Verarbeitung in Verbindung mit einem Cache eingestellt.

= VHIGH
Die Daten sollen permanent im Cache gehalten werden.

= HIGH
Die Datei soll über einen Cache bearbeitet werden.

= STD
Es wird keine besondere Anforderung bzgl. der Performance gestellt. Die Datei wird nicht über einen Cache bearbeitet (Zu Cache-Verarbeitung siehe Handbuch „Einführung in das DVS“ [1]).

IOREG = r
Nur für SAM und ISAM:
legt fest, dass die Datei im Locate-Mode verarbeitet werden soll. „r“ gibt das Register an, das die Adresse des aktuellen Satzes enthält ($2 \leq r \leq 12$).

Voreinstellung: die Datei wird im Move-Mode verarbeitet

Im Locate-Mode muss der Anwender selbst für die richtige Adressierung der Sätze im Puffer sorgen; das automatische Blocken/Entblocken der Sätze entfällt.

IOUSAGE
Gilt nur für die 31-Bit-Schnittstelle:
Mit diesem Parameter wird angegeben, wie der Cache für die Datei genutzt werden soll.

= RDWRT
Das Performanceattribut bezieht sich auf Lese- und Schreiboperationen.

= WRT
Das Performanceattribut bezieht sich auf Schreiboperationen.

= RD
Das Performanceattribut bezieht sich auf Leseoperationen.

KEYARG = relaus
Nur bei ISAM-Dateien:
gibt die Adresse eines Feldes an, das den ISAM-Schlüssel für den aktuellen Satz enthält. Dieses Feld wird ausgewertet bei den ISAM-Makros GETKY, GETFL, ELIM und SETL.

KEYLEN = länge
Nur bei ISAM-Dateien:
gibt die Länge des ISAM-Schlüssels in Byte an;
$1 \leq$ länge ≤ 255 – VALLEN – LOGLEN

Voreinstellung: KEYLEN = 8

KEYPOS = zahl

Nur bei ISAM-Dateien:

gibt die Position des ISAM-Schlüssels im Datensatz an. Bei Sätzen variabler Länge müssen 4 Byte für Satzlängen- und Steuerfeld berücksichtigt werden. Der ISAM-Schlüssel kann an beliebiger Stelle im Datensatz stehen, jedoch innerhalb einer Datei immer an derselben Position.

Voreinstellung: für Dateien mit RECFORM = V: KEYPOS = 5
für Dateien mit RECFORM = F: KEYPOS = 1

LABEL

Nur für Banddateien:

legt die Kennsatzeigenschaften für Dateien auf Magnetband oder Magnetbandkassette fest; wie die Kennsätze verarbeitet werden, hängt vom SECLEV-Operanden ab.

Voreinstellung: LABEL = (STD,1)

Für bereits bestehende Banddateien gilt immer der im VOL1-Kennsatz angegebene Normvermerk; für Ausgabedateien (OPEN OUTIN/OUTPUT) wird der LABEL-Operand ausgewertet. Enthält das Band schon Dateien oder Dateiabschnitte, wird der Normvermerk im VOL1-Kennsatz entsprechend der LABEL-Angabe versorgt/geändert (siehe auch „Hinweise zur Programmierung“ auf Seite 446).

= STD

Datei und Datenträger erhalten/haben Standardkennsätze, entsprechend DIN 66029, Austauschstufe 1.

= (STD,zahl)

Datei und Datenträger erhalten/haben Standardkennsätze entsprechend der mit „zahl“ bezeichneten Austauschstufe der DIN-Norm 66029; $0 \leq \text{zahl} \leq 3$

Auswirkungen des Operanden LABEL

	(STD,0)	**(STD,1)**	**(STD,2)**	**(STD,3)**
DIN 66029 Austauschstufe Stand	-	1 8/1972	2 6/1976	3 3/1978
Normvermerk im VOL1-Kennsatz	_ (Leerzeichen)	1	2	3
CODE=ISO-7/OWN	nicht zulässig	STD-Blöcke in Nichtstandardblöcke umgewandelt RECFORM=V: Umwandlung in D-Satzformat (nicht b. BTAM) RECSIZE > 9999 oder BLKSIZE > 9999 OPEN-Fehler	STD-Blöcke in Nichtstandardblöcke umgewandelt RECFORM=V: Umwandlung in D-Satzformat RECSIZE > 9999 oder BLKSIZE > 9999 OPEN-Fehler	STD-Blöcke in Nichtstandardblöcke umgewandelt RECFORM=V: Umwandlung in D-Satzformat RECSIZE > 9999 oder BLKSIZE > 9999 OPEN-Fehler
CODE=EBCDIC			STD-Blöcke in Nichtstandardblöcke umgewandelt	STD-Blöcke in Nichtstandardblöcke umgewandelt
Zugriffsmethode			nur SAM	nur SAM
RECFORM=U				unzulässig für Ausgabedateien; umgewandelt in (STD,2)

(STD,1) wird angenommen bei:
– RECFORM=V und CODE=EBCDIC
– BLKSIZE=STD
– FCBTYPE=PAM oder FCBTYPE=BTAM

Bei (STD,0) muss CODE=EBCDIC gelten.

Ist die Angabe im Normvermerk (VOL1-Kennsatz) kleiner als (STD,zahl), wird „zahl“ aus dem Normvermerk übernommen.

= NO
Dateikennsätze werden weder gelesen noch geschrieben (keine Dateikennsatzverarbeitung). Hat das Band Standardkennsätze, verarbeitet das System die Bandkennsätze und prüft die Zugriffsrechte.

= NSTD
Die Banddatei hat/erhält Nichtstandardkennsätze verwendet, die Dateikennsatzverarbeitung erfolgt im Benutzerprogramm. Hat der Datenträger Standardkennsätze, führt das System Bandkennsatzverarbeitung durch und prüft die Zugriffsrechte.

LARGE_FILE
Nur für Plattendateien (Zugriffsmethoden ISAM, SAM und UPAM):
Der Operand LARGE_FILE bestimmt, ob die Dateigröße 32 GB überschreiten darf oder nicht (siehe Seite 109).

= *FORBIDDEN
Die Dateigröße darf 32 GB nicht überschreiten.

= *ALLOWED
nur für 31-Bit-Schnittstellen und Dateien mit BLKCTRL ≠ PAMKEY:
Die Dateigröße darf 32 GB überschreiten.

LINK = name
Über den hier angegebenen Dateikettungsnamen („name“) stellt das DVS eine Verbindung her zur TFT und dadurch zu mit FILE-Makro definierten Datei- oder Verarbeitungseigenschaften.

„name“ darf bis zu acht Zeichen lang sein. Soll der Dateikettungsname über die Kommandoschnittstelle ansprechbar sein, muss er dem Datentyp <structured_name 1..8> entsprechen (siehe Handbuch „Kommandos“ [3]).

LOCKENV
Nur für Zugriffsmethode UPAM:
bestimmt das Lockprotokoll, das der Benutzer zur Synchronisation verwendet.

= *HOST
Shared-Update-Verarbeitung ist nur am eigenen Rechner erlaubt. Die Synchronisation erfolgt mittels Task-Lock-Manager.

= *XCS
Shared-Update-Verarbeitung soll innerhalb eines XCS-Verbunds erlaubt sein; die Synchronisation muss mit dem Distributed-Lock-Manager erfolgen.

LOGLEN = länge

Nur für ISAM-Dateien:

bestimmt die Länge einer logischen Markierung im ISAM-Index in Byte; die maximale Länge ergibt sich aus der Länge des ISAM-Schlüssels und einer evtl. vorhandenen Wertmarkierung (siehe Operand VALLEN, Seite 444), da der gesamte ISAM-Index höchstens 255 Byte lang sein darf. Es gilt also:

länge ≤ 255 – KEYLEN – VALLEN

Voreinstellung: LOGLEN=0, d.h. der ISAM-Index enthält keine logische Markierung

Der ISAM-Index kann im Anschluss an den ISAM-Schlüssel eine logische Markierung (logical Flag) enthalten, mit der Selektionskriterien bitweise dual verschlüsselt werden. In K-ISAM-Dateien werden alle logischen Markierungen eines Blocks ausgewertet und das Resultat in den nächsthöheren Indexeintrag übernommen. NK-ISAM unterstützt die Flagverarbeitung kompatibel, übernimmt die Flags jedoch nicht in den Index-Eintrag.

OPEN

gibt an, mit welchem OPEN-Modus die Datei eröffnet werden soll.

Diese Angabe kann bei Dateieröffnung durch den OPEN-Makroaufruf überschrieben werden.

Voreinstellung: OPEN = INPUT

Welche Angaben bei den Zugriffsmethoden zulässig sind, zeigt die folgende Tabelle.

OPEN-Modi beim Makro FCB

OPEN-Modus	ISAM	BTAM	SAM	UPAM
INPUT	x	x	x	x
EXTEND	x	-	x	-
INOUT	x	x	-	x
OUTIN	x	x	-	x
OUTPUT	x	x	x	-
REVERSE	-	x	x	-
SINOUT	-	x	-	-
UPDATE	-	-	x	-

x OPEN-Modus zulässig

- OPEN-Modus nicht zulässig

Die einzelnen OPEN-Modi sind unter den entsprechenden Zugriffsmethoden beschrieben.

= INPUT
Eine existente Datei (Definition siehe auch den Hinweis bei Abschnitt „Ablauf der OPEN-Verarbeitung“ im Kapitel „OPEN-Verarbeitung“ im Handbuch „Einführung in das DVS“ [1]) wird gelesen.

= EXTEND
Eine Datei wird erweitert, d.h. an das Dateiende werden weitere Datenblöcke angefügt, oder die Datei wird ab einem bestimmten Punkt überschrieben; es sind nur sequenzielle Schreiboperationen zulässig.

= INOUT
Eine existente Datei wird für nichtsequenzielle Verarbeitung eröffnet; es sind Schreib- und Leseoperationen zulässig.

= OUTIN
Die Datei wird erstellt oder – falls bereits existent – ab Dateianfang überschrieben. Es sind sowohl Schreib- als auch Leseoperationen zulässig (nichtsequenziell).

= OUTPUT
Die Datei wird sequenziell erstellt oder – falls bereits existent – ab Dateianfang überschrieben.

= REVERSE
Eine existente Datei wird als Eingabedatei für sequenzielles Lesen mit Verarbeitungsrichtung Dateiende → Dateianfang eröffnet.
Über den Operanden VSEQ im Makro FILE kann die Dateiabschnittsnummer des zu verarbeitenden Dateiabschnitts angegeben werden. Banddateien sind nach Abschluss der OPEN-Verarbeitung auf das Ende des Dateiabschnitts positioniert. Ohne VSEQ-Angabe wird der letzte Dateiabschnitt bearbeitet. Automatischer Bandwechsel wird nicht unterstützt.

= SINOUT
Nur für BTAM-Banddateien:
Die Datei muss existent sein, das Band darf nicht auf Bandanfang positioniert sein; Datenblöcke können gelesen oder geschrieben werden, es erfolgt keine Kennsatzverarbeitung.
Eine Datei, die sich über mehrere Spulen erstreckt, kann nicht mit SINOUT verarbeitet werden.

= UPDATE
Nur für SAM-Plattendateien:
die Datei soll im Locate-Mode (Ortungsbetrieb) verarbeitet werden.

OPTION = code
Mit diesem Operanden kann eine Liste von Optionen angegeben werden. Standardmäßig werden keine Codes hinterlegt.
Für „code“ kann angegeben werden: GLODEF oder NOWAIT.

= GLODEF
Wird der Dateiname ohne explizite Userid angegeben und die Datei nicht unter der Benutzerkennung des Aufrufers gefunden, wird ein zweiter Leseversuch unter der System-Standardkennung durchgeführt (siehe Abschnitt „Zugriff über die System-Standardkennung“, Handbuch „Einführung in das DVS“ [1]). Dies gilt nur, wenn zum OPEN-Zeitpunkt kein TFT-Eintrag existiert, da die TFT bereits den Pfadnamen enthält.

= NOWAIT
Läuft eine Ein-/Ausgabe auf einen gerätebedingten Fehler (z.B. Gerät INOP), wartet das Programm nicht auf eine Reaktion des Operators, sondern verzweigt sofort auf den EXLST-Ausgang ERRADDR. Der Wert NOWAIT wird nur in Zusammenhang mit der Angabe PARMOD=31 und für die Zugriffsmethoden PAM und ISAM akzeptiert.

OVERLAP = YES
Nur für ISAM-Dateien:
in Zusammenhang mit der Definition eines zweiten Ein-/Ausgabebereichs im Programm (IOAREA2 im FCB), können Leseoperationen (GET/GETR) überlappend durchgeführt werden.

Voreinstellung: OVERLAP = NO

Bei NK-ISAM bedeutet „überlappende Verarbeitung“, dass benachbarte Blöcke ebenfalls in den ISAM-Pool eingelesen werden. OVERLAP=YES sollte nur bei vorwiegend sequenziellem Lesen genutzt werden.

PAD = zahl
Nur für ISAM-Dateien, die sequenziell erstellt werden (ISAM-Makroaufruf PUT):
der „Blockfüllungsfaktor“ PAD gibt an, wie viel Platz im Datenblock für spätere Erweiterungen frei gehalten werden sollen (in Prozent der mit BLKSIZE definierten Blocklänge). Die PAD-Angabe wirkt sich somit auf die Blocksplittingrate bei nichtsequenzieller Dateierweiterung aus.

Voreinstellung: PAD = 15

Die PAD-Angabe wirkt sich unterschiedlich aus bei NK-ISAM und K-ISAM. Bei NK-ISAM wird der Block mindestens bis zur PAD-Grenze gefüllt, bei K-ISAM höchstens bis zur PAD-Grenze.

PAMREQS = zahl
Nur für UPAM-Verarbeitung:
gibt an, wie viele asynchrone Ein-/Ausgaben höchstens gleichzeitig angefordert werden können (pro PAM-Makroaufruf; siehe PAM-Makro, Operand REQNO (Seite 763);
$0 \leq$ zahl ≤ 100).

Voreinstellung: PAMREQS = 1

PAMTOUT = zahl
Nur für UPAM-Verarbeitung:
gibt an, wie lange ein Auftrag auf die angeforderten Sperren warten soll (in Sekunden).

$0 \leq$ zahl ≤ 43200

Voreinstellung: PAMTOUT = 0

Sind nach der angegebenen Zeit die Sperren nicht verfügbar, wird der EXLST-Ausgang DLOCK oder PGLOCK angesprungen. PAMTOUT=0 bedeutet, dass die Steuerung sofort zurückgegeben wird, egal, ob die Sperren verfügbar sind oder nicht.

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.

Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

Der Generierungsmodus kann für alle Makroaufrufe eines Programms mit dem Makro GPARMOD global eingestellt werden.

Der PARMOD-Operand in den DVS-Makroaufrufen setzt die Voreinstellung durch einen GPARMOD-Aufruf oder (bei fehlendem GPARMOD) die Voreinstellung des Assemblers außer Kraft.

Alle PARMOD-Angaben für eine Dateieröffnung müssen den gleichen Wert enthalten.

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst.
Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

PASS = kennwort
Ist die Datei mit Kennwörtern geschützt, muss das für den Zugriff nötige Kennwort entweder in der Kennworttabelle des Auftrags enthalten sein oder im FCB mit PASS angegeben werden.

Das Kennwort kann maximal 4 Byte lang sein; das Kennwortfeld (ID1PASS) wird linksbündig mit Nullen aufgefüllt bzw. ein längeres Kennwort von links abgeschnitten (entsprechend der Verarbeitung von Adresskonstanten in Assembler-Programmen).

Die Regeln der Kennwortangabe sowie die Hierachie der Kennwörter sind an verschiedenen Stellen im Handbuch beschrieben; hier wird auf die Beschreibung im CATAL-Makroaufruf verwiesen.

POOLLNK = name
Nur für ISAM-Dateien, die in Benutzer-ISAM-Pools verarbeitet werden (NK-ISAM):
„name“ ist der bis zu 8 Zeichen lange „Poolkettungsname“, der in die TFT eingetragen wird. Dieser Poolkettungsname muss einem ISAM-Pool zugewiesen sein (siehe Makros ADDPLNK, Seite 113 und CREPOOL, Seite 241).

Soll der Poolkettungsname über die Kommandoschnittstelle ansprechbar sein, muss er dem Datentyp <structured_name 1..8> entsprechen (siehe Handbuch „Kommandos“ [3]).

RECFORM
Gibt zum einen das Satzformat der mit „pfadname“ bezeichneten Datei an, zum anderen, welche Steuerzeichen bei der Ausgabe über einen Drucker zu berücksichtigen sind.

Voreinstellung: RECFORM = (V,N)

Das Satzformat wird bei den Zugriffsmethoden SAM und ISAM berücksichtigt.
UPAM verarbeitet Dateien nur blockweise, eine RECFORM-Angabe wird ignoriert.
BTAM ist zwar auch eine blockorientierte Zugriffsmethode, akzeptiert jedoch eine RECFORM-Angabe.

Die Satzformate sind im Kapitel „Zugriffsmethoden“ im Handbuch „Einführung in das DVS“ [1] detailliert beschrieben. Für den Zusammenhang zwischen RECFORM- und RECSIZE-Angabe siehe Operand RECSIZE; für die Drucksteuerzeichen-Auswertung wird auf das PRINT-DOCUMENT-Kommando (Operanden CONTROL-MODE und LINE-SPACING) in den Handbüchern „Kommandos“ [3] und „SPOOL“ [4] verwiesen.

Bei Band-Ausgabedateien mit RECFORM=V und CODE≠EBCDIC oder LABEL=(STD,n) (n > 1) wird der Inhalt von Satz- und Blocklängenfeld intern in das D-Format umgewandelt: der Wert für Satz-/Blocklänge wird als Dezimalzahl dargestellt. Die Blocklänge muss für solche Dateien kleiner als 10000 Byte sein. Die Format-D-Sätze werden bei der Eingabe wieder in die hexadezimale Form gebracht, bevor sie in den Bereich des Benutzers übertragen werden.

= V
„pfadname“ besteht aus Sätzen variabler Länge, d.h. der Anwender muss bei der Programmierung berücksichtigen, dass den Datensätzen ein 4 Byte langes Feld vorangestellt wird, das in den Bytes 1-2 die Satzlänge als Binärzahl enthält. Die Bytes 3-4 werden vom System genutzt. Bei Eingabedateien wird das Satzlängenfeld vom System versorgt, bei Ausgabedateien muss der Anwender das Satzlängenfeld versorgen. Eine Angabe im RECSIZE-Operanden bezeichnet die maximale Satzlänge. Für BTAM-Dateien wird die Angabe RECFORM=V wie RECFORM=U behandelt.

= F
„pfadname“ besteht aus Sätzen fester Länge, d.h. der Anwender braucht kein Satzlängen- und Steuerfeld zu berücksichtigen. Alle Sätze der Datei haben die gleiche Länge, die mit dem Operanden RECSIZE festgelegt wird. Bei BTAM bedeutet dies: alle Blöcke haben die gleiche Länge (hier ist die BLKSIZE-Angabe von Bedeutung, nicht die RECSIZE-Angabe).

= U
„pfadname“ besteht aus Sätzen „undefinierter“ Länge; jeder Datenblock enthält nur einen Satz, dessen Länge bei der Eingabe vom System, bei der Ausgabe vom Anwender in einem Register übergeben wird (siehe Operand RECSIZE). Außer bei BTAM wird bei RECFORM=U die Angabe LABEL=(STD,3) in (STD,2) umgewandelt.

Die Angabe RECFORM=U ist für ISAM-Dateien nicht zulässig.

= (...,N)
„pfadname“ ist keine Druckdatei, enthält also keine Drucksteuerzeichen und sollte nicht mit Steuerzeichenauswertung ausgedruckt werden.

= (...,M)
Das erste Datenbyte eines jeden Datensatzes wird als Steuerzeichen im EBCDI-Code interpretiert. Die Datei kann mit dem Kommando PRINT-DOCUMENT, Operand LINE-SPACING=*BY-EBCDIC-CONTROL ausgedruckt werden. Bei ISAM-Dateien wird der ISAM-Index berücksichtigt.

= (...,A)
Das erste Datenbyte eines jeden Datensatzes wird als ASA-Steuerzeichen interpretiert. Die Datei kann mit dem Kommando PRINT-DOCUMENT, Operand LINE-SPACING=*BY-ASA-CONTROL ausgedruckt werden.

RECSIZE
Gibt die Satzlänge an, abhängig vom Wert des RECFORM-Operanden.

Voreinstellung: bei RECFORM = V: RECSIZE = BLKSIZE
bei RECFORM = F: keine Voreinstellung

= länge
maximale Satzlänge in Byte.

Für RECFORM=V:
Für NK-ISAM-Dateien ist zu beachten, dass bei Ausnutzen der maximalen Satzlänge Überlaufblöcke entstehen. Für ISAM-Dateien darf die max. Satzlänge BLKSIZE nicht überschreiten.
Für SAM-Dateien darf die max. Satzlänge BLKSIZE-4 betragen, mit BLKCTRL= DATA nur max. BLKSIZE-16.
Für Banddateien ist die Wechselwirkung mit den Operanden CODE und LABEL zu beachten, siehe „Auswirkungen des Operanden LABEL“ auf Seite 432.

Für RECFORM=F (alle Sätze der Datei sind gleich lang):
Für ISAM-Dateien darf die max. Satzlänge BLKSIZE-4 nicht überschreiten.
Für SAM-Dateien darf die max. Satzlänge BLKSIZE betragen, mit BLKCTRL=DATA nur max. BLKSIZE-16.

Wird mit GET (Lesen) im Übertragungsmodus gearbeitet für Dateien mit RECFORM=V und RECSIZE=länge im Makro FCB und ist der zu lesende Satz länger als bei RECSIZE angegeben, so ist zu beachten:
– bei ISAM-Dateien wird mit der Fehlermeldung `DMS0AAD` abgebrochen
– bei SAM-Dateien wird der Satz in voller Länge in den Programmbereich übertragen, unabhängig von der Angabe bei RECSIZE.

Wird mit PUT (Schreiben) im Übertragungsmodus gearbeitet für Dateien mit RECFORM=V und RECSIZE=länge im Makro FCB und ist der zu schreibende Satz länger als bei RECSIZE angegeben, so tritt Folgendes ein:
– bei ISAM-Dateien wird der Satz in voller Länge in die Datei geschrieben, unabhängig von der Angabe bei RECSIZE.
– bei SAM-Dateien wird der Satz in voller Länge in die Datei geschrieben, unabhängig von der Angabe bei RECSIZE.

= reg
für RECFORM=U: mit dem Operanden RECSIZE muss ein Mehrzweckregister ($2 \leq reg \leq 12$) angegeben werden, das bei Ein- bzw. Ausgabe die aktuelle Satzlänge enthält. Bei der Eingabe wird das Register vom System versorgt, bei der Ausgabe muss es der Anwender versorgen.

RETPD = tage
Der Benutzer kann mit „RETPD“ eine Schutzfrist vereinbaren, während derer kein Schreibzugriff (Ändern, Löschen) möglich ist.

Voreinstellung: RETPD=0, d.h. die Datei kann jederzeit geändert/gelöscht werden.

„tage“ ist eine ganze Zahl (tage ≤ 32767). Sie gibt die Dauer der Schutzfrist in Tagen an. Ist die Schutzfrist abgelaufen, wird die Datei nicht automatisch gelöscht, es wird lediglich wieder Schreibzugriff zugelassen.

Die Schutzfrist kann auch über das Kommando MODIFY-FILE-ATTRIBUTES oder den Makro CATAL (siehe Seite 131) beeinflusst werden: die RETPD-Angabe wird dann sofort in den Katalogeintrag übernommen; für Banddateien kann CATAL nur vor der ersten Dateieröffnung genutzt werden.

SECLEV
Nur für Bandverarbeitung:
der Operand SECLEV (Security Level) bezieht sich auf den TPIGNORE-Eintrag im Benutzerkatalog-Eintrag (vgl. Kommando SHOW-USER-ATTRIBUTES). Im Dialogbetrieb wird eine SECLEV-Angabe ignoriert. Im Stapelbetrieb können dazu berechtigte Benutzer mit SECLEV festlegen, ob Fehlermeldungen unterdrückt und/oder zusätzliche Kennsatzprüfungen durchgeführt werden sollen.

= HIGH
Fehlermeldungen werden im Stapelbetrieb an der Konsole ausgegeben. Läuft der Auftrag unter einer Benutzerkennung mit der Berechtigung TPIGNORE=YES im Eintrag im Benutzerkatalog, kann der Operator die Fehlermeldung ignorieren.

= LOW
nur für den Band-/Dateieigentümer zulässig, wenn im Eintrag im Benutzerkatalog TPIGNORE=YES definiert ist: im Stapelbetrieb werden bestimmte Fehlermeldungen unterdrückt.

= (...,OPR)
Die Angabe OPR (= Overwrite Protection) veranlasst das System zusätzliche Kennsatzprüfungen durchzuführen:
– wird eine Datei innerhalb eines Bandes im Anschluss an eine bereits existierende Datei erstellt, werden die Kennsätze der vorhergehenden Datei überprüft;
– das Freigabedatum der neuen Datei darf nicht höher sein als das der vorhergehenden.

SHARUPD

Nur für ISAM- oder UPAM-Plattendateien:
gibt an, ob mehrere Aufträge gleichzeitig die Datei in einem anderen Modus als OPEN INPUT eröffnen dürfen.

= NO
Sobald die Datei von einem Auftrag mit OPEN ≠ INPUT eröffnet wird, wird sie für andere Aufträge gesperrt. Gleichzeitiger Zugriff mehrerer Aufträge auf die Datei ist nur möglich, wenn sie in allen Aufträgen als Eingabedatei verwendet wird, d.h. OPEN INPUT eröffnet ist. Ist die Datei bereits OPEN INPUT eröffnet, wird auch jeder Versuch, sie anders zu eröffnen, abgewiesen.

= YES
Nur für ISAM- oder PAM-Dateien:
die Datei kann gleichzeitig von mehreren Aufträgen bearbeitet werden; in allen Aufträgen muss jedoch SHARUPD=YES gelten, wenn Schreiben erlaubt ist. Bei UPAM kann der Anwender Datenblöcke, solange er sie verarbeitet, vor Zugriff durch andere Aufträge schützen. Bei ISAM werden diese Sperren – wenn nötig – vom System automatisch gesetzt, bei NK-ISAM als Satzsperren bzw. „Schlüsselsperren“, bei K-ISAM als Blocksperren. Bei NK-ISAM müssen Dateien, die für Shared-Update-Verarbeitung eröffnet werden, in hostspezifischen ISAM-Pools verarbeitet werden. Gleichzeitig ist für ISAM-Dateien die WROUT-Funktion eingeschaltet (siehe Operand WROUT).

= WEAK
Nur für UPAM-Verarbeitung:
garantiert Schreibsicherheit, aber nicht Lesesicherheit: nur ein Auftrag kann die Datei zur Aktualisierung eröffnen, andere Aufträge können sie aber gleichzeitig als Eingabedatei nutzen. Der Benutzer muss in seinem Programm berücksichtigen, dass sich der Inhalt der Datei ändern kann, während er sie als Eingabedatei verwendet.
Nachstehende Tabelle zeigt die verschiedenen konkurrierenden Ebenen mit den jeweils angebotenen Schutztypen:

SHARUPD-Options	Anzahl der Benutzer, die			Sicherheitstyp	
	lesen	und/oder	schreiben	READ	WRITE
YES	n	und	m	*	*
WEAK	n	und	1	-	*
NO	n	oder	1	*	*

Der Operand WEAK wird nur bei PAM-Dateien unterstützt.
Bei SHARUPD=WEAK kann eine Datei auf einem Rechner oder auf zwei verschiedenen Rechnern verarbeitet werden, die durch mehrbenutzbare private Platten (SPD = Shareable Private Disk) verbunden sind. Zu den erlaubten SHARUPD-Kombinationen siehe Kapitel „UPAM“, Handbuch „Einführung in das DVS“ [1].
Bei FCBTYPE ≠ PAM wird SHARUPD=WEAK wie SHARUPD=NO verarbeitet.

STREAM
Nur für BTAM-Banddateien:
Mit STREAM legt der Anwender fest, ob er im „Streaming-Modus“ arbeiten will. Das bedeutet einerseits, dass er sowohl Kettung von Datenblöcken als auch den MAV-Modus verwendet (die entsprechenden Angaben sind vom Benutzer zu machen!) und dass intern die einzelnen Aufträge auch wieder gekettet werden sollen. Andererseits bedeutet es, dass im Falle eines „Streaming“-Device dieser Modus hardwaremäßig eingestellt werden soll.

= NO
Streaming-Modus nicht einschalten.

= YES
Streaming-Modus einschalten.

TAPEWR
Nur für Dateien auf Magnetbandkassetten:
der Anwender kann bestimmen, ob die Ausgabe gepuffert erfolgen soll.

= DEVICE-BUFFER
Die Ausgabe wird über die Gerätesteuerung gepuffert, wodurch eine hohe Datenübertragungsrate erzielt werden kann.

= IMMEDIATE
Die Ausgabe wird nicht gepuffert.

TPMARK
Nur für Banddateien ohne Standardkennsätze:
legt fest, ob Abschnittsmarken geschrieben werden, d.h. der Operand TPMARK wird nur für Banddateien mit LABEL=NO/NSTD beim OPEN ausgewertet. Dateien mit LABEL=(STD,n) erhalten standardmäßig Abschnittsmarken nach den Kennsätzen.

= NO
Es wird keine Abschnittsmarke geschrieben.

= YES
Banddateien mit NSTD-Kennsätzen:
die Abschnittsmarke folgt dem Kennsatz.

Banddateien ohne Kennsätze:
die Abschnittsmarke wird an den Beginn des Bandes geschrieben.

TRANS
Nur für Banddateien, die als Eingabedateien genutzt werden und die nicht mit CODE=EBCDIC erstellt wurden. TRANS legt fest, wie der Code der Datei beim Lesen umgesetzt werden soll.

= YES
ISO-7-Bit-Code oder OWN-Code werden in EBCDI-Code umgesetzt.

= NO
ISO-7-Bit-Code wird mit einer führenden Null in ein 8-Bit-Format umgesetzt.

TRTADR = relaus
Der Operand gibt die Adresse der benutzereigenen Übersetzungstabelle für das Lesen von einer Banddatei an. Er darf nur in Verbindung mit CODE=OWN oder CODE null angegeben werden.

TRTADW = relaus
Der Operand gibt die Adresse der benutzereigenen Übersetzungstabelle für das Schreiben in einer Banddatei an. Er darf nur bei CODE=OWN oder CODE null angegeben werden.

VALLEN = länge
Nur für ISAM-Dateien:
legt die Länge einer Wertmarkierung (Value Flag) im ISAM-Index fest.

Voreinstellung: VALLEN = 0 (der ISAM-Index enthält keine Wertmarkierung)

länge ≤ 255 – KEYLEN – LOGLEN

Wertmarkierungen werden bei NK-ISAM (BLKCTRL=DATA) und K-ISAM (BLKCTRL= PAMKEY) unterschiedlich behandelt.

Bei K-ISAM werden sie blockweise ausgewertet und entsprechend der VALPROP-Angabe in den nächsthöheren Indexeintrag übernommen.
NK-ISAM unterstützt die Verarbeitung von Wertmarkierungen kompatibel, übernimmt die Wertmarkierungen jedoch nicht in den Index-Eintrag.

VALPROP
Nur für ISAM-Dateien in Zusammenhang mit BLKCTRL=PAMKEY d.h. für K-ISAM-Dateien (NK-ISAM ignoriert eine VALPROP-Angabe):
VALPROP (= Value Propagation) legt fest, wie die Wertmarkierung in die Indexeinträge zu übernehmen ist.

= MIN
Der niedrigste Wert für die Wertmarkierung innerhalb eines Daten- oder Indexblocks wird in den Indexeintrag der nächsthöheren Stufe übernommen.

= MAX
Der höchste Wert der Markierung im Daten-/Indexblock wird übernommen.

VARBLD = reg
Nur für SAM-Dateien mit RECFORM=V, die im Locate-Mode verarbeitet werden (siehe auch Operand IOREG):
legt das Register fest (2 ≤ reg ≤ 12), in dem das DVS den freien Platz im zu schreibenden Block anzeigt (in Byte).

WRCHK
Nur für die Verarbeitung von Plattendateien:
gibt an, ob bei Schreiboperationen „Kontroll-Lesen“ erfolgt, „WRCHK“ wird nicht in den Katalogeintrag aufgenommen und muss daher vor jeder Verarbeitung/vor jedem Öffnen wiederholt werden.

Kontroll-Lesen: Prüfung auf Aufzeichnungsfehler (→ Error-Recovery-Maßnahmen). Ist der Fehler nicht behebbar, wird der EXLST-Ausgang ERRADR angestoßen. Das Kontroll-Lesen geht wegen der zusätzlichen Plattenumdrehungen stark zulasten der Performance.

= NO
Es erfolgt kein Kontroll-Lesen.

= YES
Es erfolgt Kontroll-Lesen.

WROUT
Für ISAM-Verarbeitung:
WROUT steuert das Zurückschreiben geänderter Blöcke auf die Platte. Bei Shared-Update-Verarbeitung oder in taskübergreifenden ISAM-Pools gilt implizit WROUT=YES: geänderte Blöcke werden stets sofort auf Platte zurückgeschrieben (siehe auch „Hinweise zur Programmierung“ auf Seite 446).

Voreinstellung:	bei „normaler“ Dateiverarbeitung:	WROUT = NO
	bei Shared-Update-Verarbeitung:	WROUT = YES
	in taskübergreifenden ISAM-Pools:	WROUT = YES
	in tasklokalen ISAM-Pools, für die WROUT=YES gilt:	WROUT = YES

= NO
Der geänderte Block wird erst dann zurückgeschrieben, wenn der Inhalt des betreffenden Pufferbereichs ersetzt werden muss, spätestens beim Schließen der Datei.

= YES
Jeder geänderte Block wird sofort zurückgeschrieben, sodass die Konsistenz der Daten auf Platte und im virtuellen Speicher stets gewährleistet ist. Allerdings ist damit auch eine erhöhte Ein-/ Ausgaberate verbunden.

Rückkehr-Information (Beispiel)

Der vom OPEN im FCB-Feld ID1ECB hinterlegte Returncode 0D33 hat folgende Bedeutung, je nach den Angaben des Aufrufers im Dateinamen und im OPTION Parameter.

Dateiname	OPTION	Bedeutung von 0D33
:cat:$user.file		Die Datei ist unter der Kennung $user auf dem Pubset cat nicht vorhanden.
$user.file		Die Datei ist unter der Kennung $user auf dem Default-Pubset von $user nicht vorhanden.
$.file		Die Datei ist unter der für DEFLUID generierten Kennung auf dem dort angegebenen Pubset nicht vorhanden.
:cat:$.file		Die Datei ist unter der für DEFLUID generierten Kennung auf dem Pubset cat nicht vorhanden.
file	nicht GLODEF	Die Datei ist unter der Aufrufer-Kennung auf deren Default-Pubset nicht vorhanden.
file	GLODEF	Die Datei ist weder unter der Aufrufer-Kennung auf deren Default-Pubset noch unter der für DEFLUID generierten Kennung auf dem dort angegebenen Pubset vorhanden.
:cat:file	nicht GLODEF	Die Datei ist unter der Aufrufer-Kennung auf dem Pubset cat nicht vorhanden.
:cat:file	GLODEF	Die Datei ist weder unter der Aufrufer-Kennung noch unter der für DEFLUID generierten Kennung auf dem Pubset cat vorhanden.

Eine Modifikation des Dateinamens im Feld ID1FILE innerhalb einer OPEN-EXIT-Routine wird ignoriert.

Hinweise zur Programmierung

FCB-Aufbau

Der Aufbau eines 31-Bit-FCBs unterscheidet sich wesentlich von dem eines 24-Bit-FCBs:

– Der FCB-Makro expandiert im 31-Bit-Modus eine CSECT-Anweisung, wenn vorher noch kein weiterer DVS-Aktionsmakro im 31-Bit-Modus aufgerufen worden ist.
– Es gibt keine FCB-Erweiterung mehr, d.h. alle Daten sind im FCB selbst untergebracht.
– Die logischen Routinen der Zugriffsmethoden SAM und ISAM werden nicht mehr in den FCB eingebracht; der FCB enthält nur noch die Adressen der logischen Routinen.
– Alle 3-Byte-Adressen sind eliminiert, es wurden die entsprechenden 4-Byte-Adressen eingeführt.
– Der FCB hat eine feste einheitliche Größe.

FCB-Modifizierung

Bei Dateieröffnung prüft das DVS die FCB-Einträge und legt mit diesen Einträgen einen privilegierten Dateisteuerblock (TPR-FCB) an, eine spätere Änderung der FCB-Werte wird ignoriert. Der Dateisteuerblock kann nur dann modifiziert werden, wenn die Datei geschlossen (CLOSE) und anschließend wieder eröffnet wird (OPEN).

Wird ein Operand im FCB-Makroaufruf nicht angegeben, wird der Standardwert angenommen. Bei der Angabe eines Nullstring-Operanden wird angenommen, dass der Wert des Operanden von einem FILE-Aufruf oder vom Katalogeintrag der Datei geliefert wird (Ausnahme: Operand LINK).

Operanden für Plattendateien

FCB-Operand	Nulloperand: über Katalog versorgter Operand	BTAM	SAM	ISAM	PAM	Operand im FILE-Makro
BLIM		i	x	i	i	x
BLKCTRL	x	x	x	x	x	x
BLKSIZE	x	x	x	x	x	x
BTAMRQS		x	i	i	i	
BUFOFF	x	i	x	i	i	x
CHAINIO		x	i	i	i	x
CHKPT		i	x	i	i	x
CODE	x	x	x	i	i	x
DUPEKY		i	i	x	i	x
EXIT		x	x	x	x	
FCBTYPE	x	x	x	x	x	x
FILE		x	x	x	x	x
FORM		i	x	x	i	
FSEQ	x	x	x	i	x	x
IOAREA1		x	x	x	x	
IOAREA2		x	x	x	x	
IOPERF		i	x	x	x	x
IOREG		x	x	x	i	
IOUSAGE		i	x	x	x	x
KEYARG		i	i	x	i	
KEYLEN	x	i	i	x	i	x
KEYPOS	x	i	i	x	i	x

FCB-Operand	Nulloperand: über Katalog versorgter Operand	BTAM	SAM	ISAM	PAM	Operand im FILE-Makro
LABEL		x	x	i	x	x
LINK		x	x	x	x	x
LARGE_FILE		i	x	x	x	x
LOCKENV		i	x	x	x	x
LOGLEN	x	i	i	x	i	x
OPEN		x	x	x	x	x
OPTION		x	x	x	x	x
OVERLAP		i	i	x	i	x
PAD		i	i	x	i	x
PAMREQS		i	i	i	x	
PAMTOUT		i	i	i	x	
PARMOD		x	x	x	x	
PASS		x	x	x	x	
POOLLNK		i	i	x	i	x
RECFORM	x	x	x	x	i	x
RECSIZE	x	x	x	x	i	x
RETPD		x	x	x	x	x
SECLEV		x	x	i	x	
SHARUPD		i	x	x	x	x
STREAM		x	x	i	i	
TAPEWR		x	x	i	x	x
TPMARK		x	x	i	x	x
TRANS		x	x	i	i	x
TRTADR		x	x	i	i	
TRTADW		x	x	i	i	
VALLEN	x	i	i	x	i	x
VALPROP	x	i	i	x		x
VARBLD	x	i	x	x	i	
WROUT		i	i	x	i	x
WRCHK		i	x	x	x	x

Legende

i Operanden werden ignoriert

x Operanden können angegeben werden

Operand BLKSIZE

Der Anwender muss/kann BLKSIZE=(STD,n) angeben, wenn

– der Datensatz länger ist als 2048 Byte oder

– die Satzlänge unwirtschaftlich ist für eine Blocklänge von 2048 Byte. Wenn der Benutzer Sätze fester Länge (RECSIZE=F) zu 1500 Byte hat und BLKSIZE=STD definiert hat, so würden pro Block 548 Byte verschwendet werden. Würde der Benutzer hingegen BLKSIZE=(STD,3) angeben, so würden von den 6144 Byte (3 x 2048) 6000 Byte benutzt werden und bei drei Blöcken nur 144 Byte verschwendet werden. Die Verwendung sehr großer Blocklängen führt jedoch zu erhöhter Paging-Rate.

Operanden IOAREA1/2

Wenn eine Datei eröffnet wird, werden die Adressen der IOAREAs (falls erforderlich) erstellt und überprüft. Sie werden dann in den Systemspeicher übertragen. Demzufolge werden alle Änderungen, die diese Adressen betreffen, im FCB vollständig ignoriert. Neue Adressen werden nur wirksam nach CLOSE- und anschließendem OPEN-Makroaufruf.

Diese Art der Verarbeitung wurde gewählt, um die interne Systemverarbeitungszeit (Overhead) gering zu halten, da die Adressen der IOAREAs nicht vor jedem Aktionsmakroaufruf überprüft werden müssen. PAM und BTAM lassen die Definition der Pufferadressen in ihren Aktionsmakroaufrufen zu. Diese Pufferadressen werden bei Ausgabe der Aktionsmakroaufrufe überprüft.

Wird eine SAM-Datei im UPDATE-Modus eröffnet, so wird der Puffer IOAREA2 nicht verwendet.

Ist IOAREA1=NO angegeben, so muss der Wert für IOAREA2 ebenfalls NO sein. Ist IOAREA1 mit relaus definiert, muss der Wert für IOAREA2 ebenfalls „relaus“ oder NO sein. Ist IOAREA1 nicht angegeben, so darf IOAREA2 ebenfalls nicht angegeben werden oder muss mit NO definiert sein.

Operand LABEL

Für eine Datei, die INPUT, INOUT, EXTEND oder REVERSE eröffnet wird, ignoriert das System LABEL=(STD,n) und bezieht sich auf die im Bandkennsatz (VOL1) angegebene Austauschstufe (Normvermerk).

Für eine Ausgabedatei ist die hier angegebene Austauschstufe maßgebend. Wurde nur STD angegeben, ist die Austauschstufe 3 maßgebend.

Enthält das zugewiesene Band schon eine oder mehrere Dateien bzw. Dateiabschnitte, dann muss die Austauschstufe mit dem Normvermerk im ersten Bandkennsatz übereinstimmen.

Andernfalls wird der Normvermerk entsprechend versorgt:

Austauschstufe	0	1	2	3
Normvermerk	Leerzeichen	1	2	3

Einschränkungen

- Austauschstufe 1: außer bei BTAM werden bei CODE=ISO7/OWN Standardblockangaben in Nichtstandardblockangaben und V-Sätze (RECFORM=V) in D-Sätze konvertiert. Ist dies nicht möglich, z.B. wenn RECSIZE/BLKSIZE > 9999, erfolgt ein OPEN-Fehler.
- Austauschstufe 2: Nur SAM-Dateien dürfen verarbeitet werden. STD-Blockangaben werden wie bei (STD,1) konvertiert, wobei hier auch Bänder im EBCDIC davon betroffen sind.
- Austauschstufe 3: Die Angabe RECFORM=U ist für Ausgabedateien unzulässig.

Regeln

- Aus LABEL=(STD,3) wird (STD,1) bei FSEQ=0/1.
- bei RECFORM=V und CODE=EBCDIC gilt implizit LABEL=(STD,1).
- bei BLKSIZE=STD gilt implizit LABEL=(STD,1).
- bei BLKSIZE=länge und RECFORM=U gilt implizit LABEL=(STD,2).
- Hat ein Band schon eine bzw. mehrere Dateien bzw. Dateiabschnitte und ist der Normvermerk im ersten Bandkennsatz kleiner als die implizit angenommene DIN-Austauschstufe, wird die Versionsnummer dem Bandkennsatz entnommen.

Operand WROUT

Die Funktion WROUT erhöht die Sicherheit der ISAM-Dateibearbeitung. Bei einem Systemzusammenbruch sind dann nur die vom letzten Makroaufruf bearbeiteten Sätze fehlerhaft bzw. gehen verloren (Ausnahme: beim Makroaufruf PUT werden nur Blöcke geschrieben).

Die erhöhte Anzahl von Ein-/Ausgabe-Operationen verringert den Durchsatz.

Die Funktion WROUT wirkt nach den ISAM-Aktionsmakroaufrufen STORE, ELIM, INSRT und PUTX.

ISAM SHARED UPDATE enthält die Funktion WROUT; hier ist der Operand ohne Bedeutung.

Eine Meldung erfolgt immer bei einem von „YES“ oder „NO“ abweichenden Wert.

Der Wert im FILE-Aufruf hat Priorität vor dem Wert im FCB-Makroaufruf. Nur wenn im FILE-Aufruf der Operand nicht angegeben ist bzw. der Operandenwert fehlt (d.h. WROUT=,...), ist der im FCB-Makroaufruf angegebene Wert maßgeblich.

FCBAD – FCB-Adressen anlegen

Makrotyp: O-Typ

Für alle im Programm folgenden DVS-Makros wird der Code so generiert, dass die FCBs bei symbolischer Adressierung außerhalb der Basisregister stehen können. Die Adressen der FCBs werden dabei im Literal-Bereich abgelegt. Der FDBAD-Makro wurde zur Erleichterung der Umstellung auf BS2000 geschaffen.

Format

Operation	Operanden
FCBAD	

FEOV – Band abschließen

Der FEOV-Makroaufruf bewirkt, dass Bandwechsel eingeleitet und die Verarbeitung auf dem Folgeband fortgesetzt wird. Der Makroaufruf wird bei Banddateien, die mit OPEN REVERSE eröffnet wurden, ignoriert.

Ist bei Eingabedateien das Dateiende auf dem Band, erkennt das DVS „Dateiende“ und aktiviert die EOFADDR-Routine (siehe bei Makro EXLST Seite 397). Enthält das Band nicht das Dateiende und ist kein Folgeband zugewiesen, aktiviert das DVS die NODEV-Routine (siehe bei Makro EXLST Seite 401).

Format

Operation	Operanden
FEOV	{ fcbadr / (1) } [,PARMOD={ 24 / 31 }]

Operandenbeschreibung

fcbadr
Adresse des FCB der zu verarbeitenden Datei.

(1)
Register 1 entält die FCB-Adresse.

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.

Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst.
Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

Hinweis zur Programmierung

Der FEOV-Makroaufruf zerstört die Register 0, 1, 14 und 15.

Beispiel für Bandwechsel bei einer SAM-Datei

Mit FEOV-Makroaufruf wird Bandwechsel ausgelöst.

```
FEOVTEST START
         LDBASE 3
         USING  *,3
         .
         .
         FILE   TAPE.TEST,LINK=AUS,DEVICE=T9P,VOLUME=(C1776A,C2921A)
         .
         .
         OPEN  TAPE,OUTPUT          BANDDATEI 'TAPE' EROEFFNEN
         .
         .
         PUT   TAPE,RECOUT          SATZ SCHREIBEN
         .
         .
         FEOV  TAPE                 BANDWECHSEL EINLEITEN
         .
         .
         CLOSE TAPE                 BANDDATEI 'TAPE' SCHLIESSEN
*
ENDE     TERM
*
TAPEND   LR    8,0                  KENNSATZROUTINE
         MVC   0(L'BEGIN,8),BEGIN   *
         LBRET TAPE,1               *
*
TAPE     FCB   FCBTYPE=SAM,BLKSIZE=(STD,3),LINK=AUS,RECFORM=F,        -
               RECSIZE=22,EXIT=TAPEXIT
*
TAPEXIT  EXLST COMMON=ENDE,LABEOV=TAPEND
*
RECOUT   DS    CL22
*
BEGIN    DC    CL80'UTL1 BENUTZERETIKETT'
         .
         .
         END
```

FILE – Dateimerkmale definieren/Dateiverarbeitung steuern

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form) (siehe Seite 870)

Der FILE-Makroaufruf bearbeitet permanente und temporäre Dateien (nicht EAM-Dateien) sowie Dateigenerationen. Er kann verwendet werden, um neue Dateien bzw. Katalogeinträge zu erstellen, Dateieigenschaften zu verändern sowie Dateien von privaten Datenträgern zu übernehmen.

Mit FILE können außer einer Schutzfrist (Operand RETPD) keine Dateischutzmerkmale wie Kennwörter, Zugriffsart usw. definiert oder verändert werden. Beim Erstellen eines Katalogeintrags mit FILE werden die entsprechenden Voreinstellungen des Systems übernommen. Anschließend können diese Werte mit einem CATAL-Makroaufruf verändert werden.

Über die Task File Table (TFT) stellt der FILE-Makroaufruf eine Verbindung her zwischen Programm und Datei, zwischen den im FILE-Makroaufruf bzw. im Katalogeintrag definierten Dateimerkmalen und dem FCB-Makroaufruf.

Hauptfunktionen des FILE-Makroaufrufs

- Katalogeintrag erstellen für neue Dateien/Dateigenerationen
- Geräte und Datenträger anfordern
- Speicherplatz zuweisen oder freigeben
- TFT-Eintrag erstellen mit Angaben zur Dateiverarbeitung (Datenstruktur, Öffnungsmodus usw.)
- Datenorganisation auf Bändern definieren

Im Anschluss an diese Einleitung finden Sie eine Übersicht über die Funktionen des FILE-Makroaufrufs auf Operandenebene. Die einzelnen Themen (wie z.B. TFT, TST usw.) sind im Einführungsteil, d.h. in den ersten Kapiteln dieses Handbuches detailliert beschrieben.

Katalogeintrag

Ist die im FILE-Makroaufruf angegebene Datei/Dateigeneration noch nicht katalogisiert, wird ein Katalogeintrag erstellt. Ist sie bereits katalogisiert, greift das DVS bei Dateieröffnung auf den Katalogeintrag zu und aktualisiert ihn gegebenenfalls beim Schließen der Datei. Angaben zu den Operanden IOPERF, IOUSAGE, DEVICE, VOLUME, SPACE, DDEVICE, DVOLUME, DSPACE, STATE=FOREIGN (für Banddatei) und FSEQ (nur zum Teil) werden ausgewertet und in den Katalogeintrag übernommen, ansonsten werden die entsprechenden Voreinstellungen des Systems gesetzt. Angaben zu den übrigen Operanden im FILE-Aufruf werden nur im Zusammenhang mit einem Dateikettungsnamen ausgewertet und in den TFT-Eintrag übernommen.

Enthält der Katalogeintrag eine BASIC-ACL, ein DELDATE oder GUARDS, so kann mit FILE keine Banddatei erzeugt werden. (Ein FILE-Makro mit entsprechendem DEVICE-Operanden wird abgewiesen.)

Ist eine neu zu katalogisierende Datei/Dateigeneration auf privater Platte gespeichert, entnimmt das DVS die Werte für den Katalogeintrag dem F1-Kennsatz des ersten Datenträgers der Datei.

Beim Katalogisieren einer neuen Datei wird Folgendes gesetzt:

für BACKUP:	E für temporäre Datei und Arbeitsdatei; sonst: gemäß Klasse-2-Systemparameter BACKUP
für MIGRATE:	FORBIDDEN bei Angabe einer zu einem SM-Pubset gehörenden Platte im Operanden VOLUME; sonst: INHIBITED für temporäre Datei, ALLOWED für permanente Datei
für CCS:	Zeichensatz (Coded Character Set) aus dem Benutzerkatalogeintrag des Dateieigentümers. Wenn dieser Zeichensatz gleich `EDF03IRV` ist, wird kein Zeichensatz eingetragen.

Falls eine Datei einen Katalogeintrag, jedoch noch keinen Plattenspeicher hat und durch den FILE-Aufruf Plattenspeicher auf einer privaten Platte erhalten soll, darf die Datei nicht verschlüsselt sein.

Wenn eine Datei einen Katalogeintrag, jedoch noch keinen Bandtyp hat und durch den FILE-Aufruf einen Bandtyp erhalten soll, so darf die Datei nur dann verschlüsselt sein, wenn es sich um eine Dateigeneration handelt. Diese wird dann entschlüsselt.

Beim Katalogisieren einer neuen Dateigeneration werden die Verschlüsselungsattribute vom Gruppeneintrag in den neuen Katalogeintrag übernommen. Dies gilt nicht für Dateigenerationen auf Band.

Dateikettungsname / Task File Table (TFT)

Ist im FILE-Makroaufruf mit dem LINK-Operanden ein Dateikettungsname angegeben, erzeugt das System einen Eintrag in der auftragsbezogenen TFT. In diesen TFT-Eintrag werden im aktuellen FILE-Makroaufruf angegebene Werte übernommen, auch NULL-Operanden werden berücksichtigt (siehe unten). Die Werte in der TFT werden bei Dateieröffnung in den Dateisteuerblock (FCB) übernommen.
Zum OPEN-Zeitpunkt befinden sich dann die Angaben bzgl. einer Datei im:

- TFT-Eintrag
- Dateisteuerblock (FCB) des Programms
- Katalogeintrag der Datei

Aus dem Inhalt des Dateisteuerblocks wird später der Katalogeintrag aktualisiert (siehe OPEN- und CLOSE-Verarbeitung, Handbuch „Einführung in das DVS“ [1]).

Poolkettungsname / ISAM-Pools

Mit NK-ISAM werden ISAM-Dateien in ISAM-Pools verarbeitet. Die Verbindung zwischen Benutzer-ISAM-Pool und Datei wird über den Poolkettungsnamen hergestellt, der mit dem Operanden POOLLNK angegeben wird. Ohne Angabe eines Poolkettungsnamens wird die Datei in einem Standardpool des Systems bearbeitet, falls sie mit SHARUPD=NO geöffnet wird; andernfalls kommt eine Fehlermeldung mit dem Fehlerschlüssel DMS0D9B.

Zugriffsmethoden

Mit dem FILE-Makroaufruf können – je nach Zugriffsmethode – Datenstrukturen definiert werden, z.B. Satzlänge, Blocklänge, usw.

Auf Besonderheiten und Wechselwirkungen der Operanden wird in der Operandenbeschreibung hingewiesen.

NULL-Operanden

Einige Operanden des FILE-Makros können in Verbindung mit einem Dateikettungsnamen als so genannte „NULL-Operanden“ angegeben werden. Dies bedeutet, dass bei dem entsprechenden Operanden kein Operandenwert angegeben wird (leere Zeichenfolge als Operandenwert).

Bei Dateieröffnung werden die Informationen für diese Dateimerkmale aus dem Katalogeintrag in den Dateisteuerblock (FCB) übernommen.

```
FILE datei,LINK=name,FCBTYPE=,RECFORM=,...
```

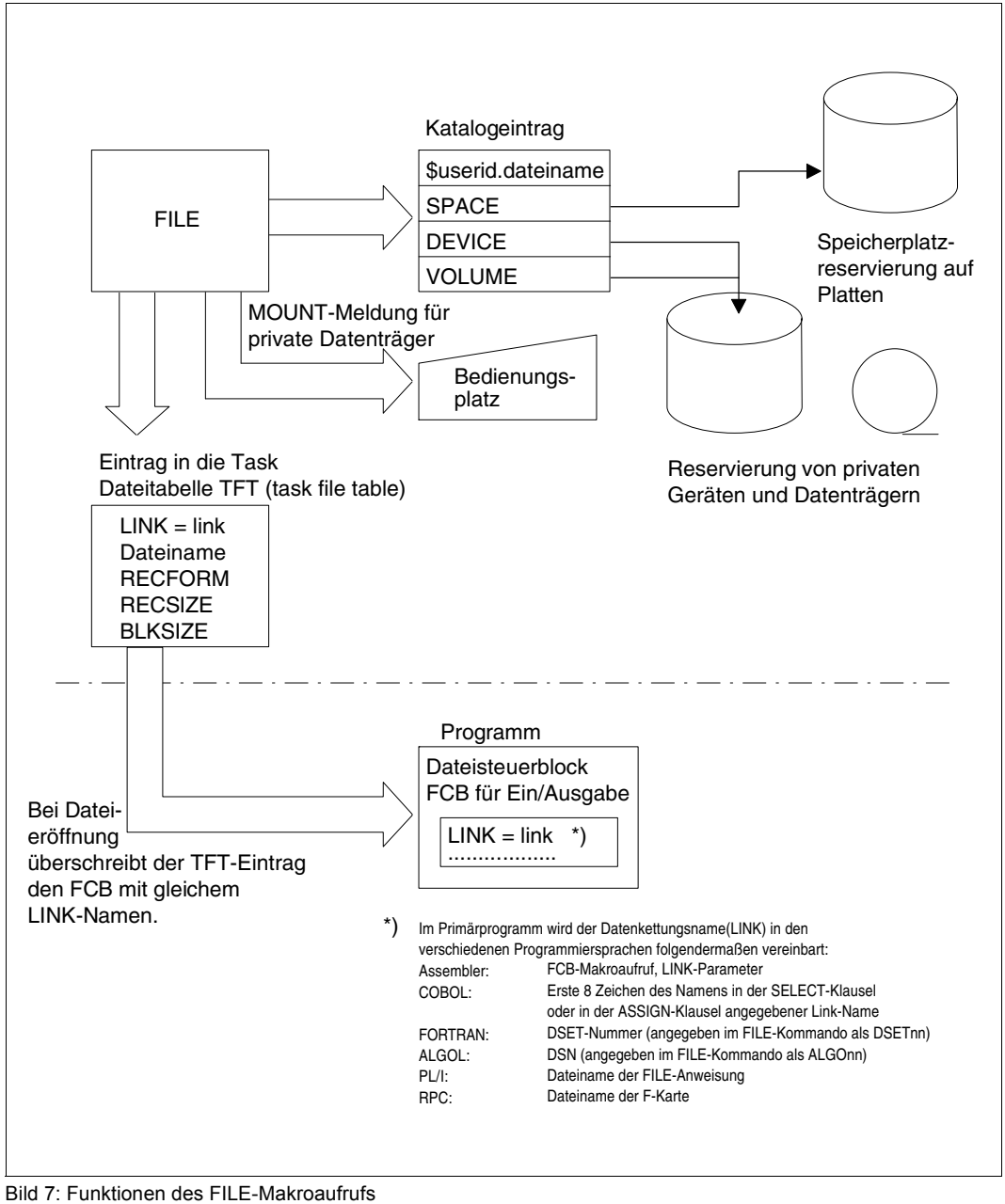
Folgende Operanden können in Verbindung mit einem Dateikettungsnamen als NULL-Operanden angegeben werden:

BLKCTRL, BLKSIZE, BUFOFF, CODE, FCBTYPE, FSEQ, IOPERF, IOUSAGE, KEYLEN, KEYPOS, LOGLEN, RECFORM, RECSIZE, VALLEN und VALPROP

Der Begriff „NULL-Operand“ ist nur für die o.g. Operanden des Makroaufrufes FILE zulässig. Auch bei anderen Makros ist es möglich, keinen Operandenwert anzugeben; dies führt dort aber in der Regel zur Übernahme der Voreinstellung bzw. des Standardwertes.

Version des FILE-Makroaufrufs

Mit dem Operanden VERSION wird gesteuert, welches Aufrufformat generiert wird. Ohne Angabe von VERSION bzw. bei Angabe von VERSION=0 werden Operandenliste und SVC für das alte Format (Stand BS2000 V9.0) erzeugt. Die zu BS2000 V9.5 neu eingeführten Operanden und die ab BS2000 V9.5 gültigen Gerätetypen werden ab VERSION=1 unterstützt. Neuerungen der BS2000-Versionen V10.0 bzw. BS2000/OSD-BC V1.0 – insbesondere das neue Layout der Operandenliste mit Auslagerung der variablen Teile in eigene Listen (siehe „Hinweise zur Programmierung“ auf Seite 513) – können ab VERSION=2 genutzt werden.



*) Im Primärprogramm wird der Datenkettungsname(LINK) in den verschiedenen Programmiersprachen folgendermaßen vereinbart:

Assembler:	FCB-Makroaufruf, LINK-Parameter
COBOL:	Erste 8 Zeichen des Namens in der SELECT-Klausel oder in der ASSIGN-Klausel angegebener Link-Name
FORTRAN:	DSET-Nummer (angegeben im FILE-Kommando als DSETnn)
ALGOL:	DSN (angegeben im FILE-Kommando als ALGOnn)
PL/I:	Dateiname der FILE-Anweisung
RPC:	Dateiname der F-Karte

Bild 7: Funktionen des FILE-Makroaufrufs

Funktionsübersicht

Operand	Operandenwert	Funktion / Bedeutung
Datei benennen, katalogisieren, Kettungsnamen definieren		
pfadname		– Katalogeintrag erstellen – Speicherplatz zuweisen (Primärzuweisung) – Datei/Katalogeintrag benennen, auf den sich die weiteren Operationen beziehen
DATATTR		gibt an, von welcher Referenzdatei beim Erzeugen eines TFT-Eintrags bestimmte Werte übernommen werden sollen, die nicht mit den jeweil. Operanden explizit angegeben werden
*DUMMY		Pseudodatei (DUMMY-Datei) definieren
LINK	name	Dateikettungsnamen definieren, für den ein TFT-Eintrag erstellt wird
POOLLNK	name	für NK-ISAM-Dateien: Poolkettungsnamen für den Benutzer-ISAM-Pool definieren
STATE	FOREIGN	Datei von privaten Datenträgern oder von einem Net-Storage-Volume importieren
Dateieigenschaften		
AVAIL	HIGH	Anforderungen bzgl. Ausfallsicherheit festlegen
BLKCTRL	PAMKEY/DATA/ DATA2K/DATA4K/ NO	Dateiformat festlegen
BLKSIZE	(STD,n) länge	Blocklänge als Vielfaches der PAM-Seite Nichtstandardblöcke (Banddateien)
CODE	EBCDIC/ISO7 ISO7D/ OWN	Banddateien: Code
EXC32GB	ALLOWED/ FORBIDDEN	Dateigröße für Plattendateien > 32 GB erlauben
FCBTYPE	ISAM/PAM/SAM BTAM	Zugriffsmethode für die Datei
KEYLEN	zahl	ISAM-Dateien: Länge des ISAM-Schlüssels
KEYPOS	zahl	ISAM-Dateien: Position des ISAM-Schlüssels
LOGLEN	zahl	ISAM-Dateien: Länge der logischen Markierung
NFTYPE	BS2000/ NODE-FILE	Dateityp für Datei auf Net-Storage: BS2000-Datei oder Node-File
RECFORM	V/F/U N/M/A	Satzformat: variabel/fest/undefiniert legt fest, ob Drucksteuerzeichen zu berücksichtigen sind

Operand	Operandenwert	Funktion / Bedeutung
RECSIZE	länge r	Satzlänge für RECFORM=F Register, das bei RECFORM=U die aktuelle Satzlänge enthält
RETPD	tage	Schutzfrist für Datei
STOCLAS		Datei beim Anlegen auf einem SM-Pubset eine Storage-Klasse zuweisen
VALLEN	zahl	ISAM-Dateien: Länge der Wertmarkierung
WORKFIL		festlegen, ob die Datei auf einem Volume-Set für Arbeitsdateien oder auf einem Volume-Set mit permanenter Datenhaltung angelegt werden soll
Geräte und Datenträger anfordern		
DDEVICE	gerät	ISAM-Dateien: Gerätetyp für Datenteil bei Index-/Datentrennung
DEVICE	gerät/WORK	Gerätetyp definieren/Arbeitsband anfordern
DVOLUME	(vsn,...)	ISAM-Dateien: Privatplatte für Datenteil bei Index-/Datentrennung
FSEQ	UNK/NEW/zahl	Banddateien: in einer Dateimenge positionieren
MOUNT	(zahl,...)	Montieraufforderung für private Datenträger
TSET	(name,vsn)	Banddateien: definiert eine Bandmenge für die Erweiterung von Dateien oder Dateimengen
TVSN	(vsn,...)	Banddateien: temporäre Datenträgerliste für die aktuelle Verarbeitung
VOLSET		festlegen, auf welchem Volume-Set eines SM-Pubsets die Datei angelegt werden soll
VOLUME	(PRIVATE,n) (vsn,...)	private Datenträger anfordern Datenträgerliste definieren
VSEQ	(L=(zahl,...))	Banddateien: bezeichnet den gesuchten Dateiabschnitt
Speicherplatzverwaltung für Plattendateien		
DSPACE	primär (primär,sekundär) (seite,zahl,ABS)	ISAM-Dateien: Speicherplatzverwaltung für den Datenteil bei Index-/Datentrennung (siehe Operand DSPACE, Seite 480)
SPACE	primär (primär,sekundär) (primär,sekundär, *KEEP) (seite,zahl,ABS)	Plattendateien: Speicherplatzzuweisung oder Speicherplatzfreigabe Absolutzuweisung

Operand	Operandenwert	Funktion / Bedeutung
Öffnungsmodus, Verarbeitungseigenschaften		
BLIM	zahl	Banddateien: Maximalzahl logischer Blöcke pro Band
BUFOFF	L/länge	Banddateien: Länge der Pufferverschiebung
BYPASS	LP/(LP,n)/ (LP,+n)/(LP,-n)	Banddateien: Kennsatzprüfung ausschalten
CHAINIO	zahl	Banddateien: Kettungsfaktor
CHKPT	NO/ANY/BLIM/ FEOV	Banddateien: automatisches Fixpunktschreiben
CLOSE	RWD/INVAL/ REPOS/DISCON/ LEAVE/KEEP-DATA-IN-CACHE	Band-/Plattendateien: Schließungsmodus für die Datei
CLOSMSG	NO/YES	SAM-Dateien: Meldungsausgabe nach Abschluss der CLOSE-Verarbeitung
DESTOC	NO/YES	Banddateien: Überschreiben von Restdaten
DISKWR	BY-CLOSE/ IMMEDIATE	Zeitpunkt nach einer Schreiboperation festlegen, zu dem die Daten der Datei sich in einem konsistenten Zustand befinden müssen
DUPEKY	YES/NO	ISAM-Dateien: Mehrfachschlüssel zulässig
IOPERF	HIGH/STD/USER-MAX/VERY-HIGH	Performance-Attribut der Datei festlegen
IOUSAGE	RD-WRT/READ/WRITE	gibt an, auf welche I/O-Operationen sich das Performance-Attribut (IOPERF) der Datei bezieht
LABEL	(STD,zahl) NO NSTD	Banddateien: Datei mit Standardkennsätzen (entsprechend DIN 66029) Banddateien ohne Dateikennsätze Banddateien mit Nichtstandardkennsätzen
LOCKENV	HOST/XCS	festlegen, ob die Datei gleichzeitig von verschiedenen Systemen aus zum Schreiben geöffnet sein kann
OPEN	INPUT/OUTPUT/ EXTEND/INOUT/ OUTIN/UPDATE/ SINOUT/REVERSE	Öffnungsmodus für die Datei
OVERLAP	YES/NO	ISAM-Dateien: überlappende Verarbeitung
PAD	zahl	ISAM-Dateien: Blockfüllung bei sequenzieller Dateierstellung
POOLSIZ	Zahl	ISAM-Dateien: Größe des dateispezifischen ISAM-Pools

Operand	Operandenwert	Funktion / Bedeutung
SECLEV	HIGH/LOW	Banddateien: Sicherheitsgrad
SHARUPD	YES/NO/WEAK	ISAM-Dateien: Shared-Update-Verarbeitung – zulässig/nicht zulässig PAM-Dateien: Shared-Update-Verarbeitung – zulässig/nicht zulässig/Schreibgarantie
STREAM	NO/YES	BTAM-Banddateien: Ein/Ausgaben im Streaming-Modus ermöglichen
TAPEWR	DEVICE-BUFFER/ IMMEDIATE	Dateien auf Magnetbandkassetten: gepufferte/ungepufferte Ausgabe
TPMARK	YES/NO	Banddateien: Abschnittsmarken schreiben
TRANS	YES/NO	Nicht-EBDIC-Banddateien umsetzen
VALPROP	MIN/MAX	K-ISAM-Dateien: Auswertung der Wertmarkierungen steuern
WRCHK	NO/YES	Plattendateien: Kontroll-Lesen bzgl. Aufzeichnungsfehler
WROUT	NO/YES	ISAM-Dateien: sofortiges Zurückschreiben geänderter Blöcke
Steuerung der Makrogenerierung		
MF	D / C / L / E	Operandenlisten / Funktionsaufruf
PREFIX		Präfix für Namen in Operandenlisten
VERSION	0 / 1 / 2 / 3	Versionseinstellung für den Makroaufruf

Format

Operation	Operanden
FILE	VERSION = 0 / <integer 1..3> ,MF = C / D / L / S / (E,<name>) / (E,(<reg 0..15>)) ,PREFIX = I / * / <name 1..1> ,<pfadname 1..54> / *DUMMY ,AVAIL = HIGH ,BLIM = <integer 1..999999> ,BLKCTRL = *BY-PROG / < > / NO / PAMKEY / DATA / DATA4K / DATA2K ,BLKSIZE = *BY-PROG / < > / STD / <integer 1..32767> / (STD,<integer 1..16>) ,BUFOFF = *BY-PROG / < > / L / <integer 0..99> ,BYPASS = LP / (LP,<integer -127..32767>) ,CHAINIO = <integer 1..100> ,CHKPT = (NO / BLIM / FEOV / ANY , ACTIVE / DUMMY) ,CLOSE = RWD / REPOS / DISCON / LEAVE / INVAL / KEEP-DATA-IN-CACHE ,CLOSMSG = NO / YES ,CODE = *BY-PROG / < > / EBCDIC / ISO7 / ISO7D / OWN ,DATATTR = (*FROM-FILE,<c-string: filename 1..54>) ,DDEVICE = <name 1..8> ,DESTOC = NO / YES ,DEVICE = <name 1..8> ,DISKWR = IMMEDIATE / BY-CLOSE

(Teil 1 von 4)

Operation	Operanden
	,DSPACE = <integer 0..2147483647> / (<integer 0..2147483647> [,<integer 0..32767>]) / (<integer 0..2147483647>,<integer 0..2147483647>,ABS) ,DUPEKY = NO / YES ,DVOLUME = <@adr> / PRIVATE / (PRIVATE,<integer 1..9>) / list-poss(255): <name 1..6> ,EXC32GB = FORBIDDEN / ALLOWED ,FCBTYPE = *BY-PROG / < > / SAM / ISAM / BTAM / PAM ,FSEQ = < > / UNK / NEW / <integer 0..9999> ,IOPERF = < > / STD / HIGH / VERY-HIGH / USER-MAX ,IOUSAGE = < > / READ / WRITE / RDWRT ,KEYLEN = *BY-PROG / < > / <integer 1..255> ,KEYPOS = *BY-PROG / < > / <integer 1..32767> ,LABEL = *BY-PROG / STD / NO / NSTD / (STD,<integer 0..3>) ,LINK = <name 1..8> ,LOCKENV = HOST / XCS ,LOGLEN = *BY-PROG / < > / <integer 0..255> ,MOUNT = 0 / <@adr> / list-poss(255): <integer 1..255> ,NFTYPE = BS2000 / NODE-FILE ,OPEN = INPUT / OUTPUT / OUTIN / INOUT / SINOUT / EXTEND / REVERSE ,OVERLAP = NO / YES ,PAD = <integer 0..99> ,POOLLNK = <name 1..8> ,POOLSIZ = <integer 128..1048576>

(Teil 2 von 4)

Operation	Operanden
	,RECFORM = *BY-PROG / < > / F / V / U / (F / V / U , N / M / A) ,RECSIZE = *BY-PROG / < > / <integer 0..32768> / <reg 2..12> ,RETPD = <integer 0..32767> ,SECLEV = HIGH / LOW / (HIGH / LOW , OPR) ,SHARUPD = NO / YES / WEAK ,SPACE = <integer -2147483647..2147483647> / (<integer -2147483647..2147483647>,<integer 0..32767>) / (<integer -2147483647..2147483647>,*KEEP) / (<integer 02147483647..2147483647>,<integer 0..32767>,*KEEP) / (<integer 1..2147483647>,<integer 1..2147483647>,ABS) ,STATE = FOREIGN ,STOCLAS = *NONE / <c-string: name 1..8> ,STREAM = NO / YES ,TAPEWR = DEVICE-BUFFER / IMMEDIATE ,TPMARK = NO / YES ,TRANS = YES / NO ,TSET = <name 1..4> / (<name1..4>,<name 1..6>) ,TVSN = <@adr> / list-poss(255) <name 1..6> ,VALLEN = *BY-PROG / < > / <integer 0..255> ,VALPROP = *BY-PROG / < > / MIN / MAX ,VOLSET = <c-string: catid 1..4> / *CONTROL ,VOLUME = <@adr> / REMOVE-UNUSED / PRIVATE / (PRIVATE [,<integer 1..9>]) / list-poss(255): <name 1..6> ,VSEQ = <@adr> / <integer 1..255> / (L=list-poss(255): <integer 1..255>)

(Teil 3 von 4)

Operation	Operanden
	,WORKFIL = NO / YES ,WRCHK = NO / YES ,WROUT = NO / YES

(Teil 4 von 4)

Voreinstellung der Operanden

Für die FILE-Operanden, bei denen keine Voreinstellung explizit beschrieben wird und zu denen ein entsprechender FCB-Operand existiert, gilt Folgendes: Ist der Operand im FILE-Makroaufruf nicht angegeben, wird dies im TFT-Eintrag vermerkt. Der für die Dateiverarbeitung gültige Wert des Operanden ergibt sich dann aus den Angaben im FCB-Makroaufruf. Ist der Operand auch im FCB-Makroaufruf nicht angegeben, wird die Voreinstellung des FCB-Makros wirksam, sofern sie nicht von OPEN durch den entsprechenden Wert aus dem Katalogeintrag oder einen anderen Wert überschrieben wird.

Operandenbeschreibung

Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.
In allen Makroaufrufen, die sich durch den MF-Operanden unterscheiden (MF=L/E/D/C), muss der Versionsoperand den gleichen Wert haben.

pfadname
bezeichnet den Pfadnamen der Datei(en) oder Dateigeneration(en) mit:
<c-string 1..54: filename 1..54>

„pfadname“ darf keine Dateigenerationsgruppe sein.

Ist „pfadname“ noch nicht katalogisiert, wird ein Katalogeintrag erstellt und der Datei entsprechend der Primärzuweisung (siehe Operand SPACE, Seite 497) Speicherplatz zugewiesen. Wird eine temporäre Datei angegeben, so muss sie von der aufrufenden Task angelegt worden sein.

Folgende Elemente können NICHT auf einem Net-Storage-Volume angelegt werden:
– Dateien mit PAM-Key
– Dateigenerationen
– Arbeitsdateien
– temporäre Dateien

Pfadname bedeutet [:catid:][$userid.]dateiname

catid
Katalogkennung;
Default-Catid: die der Userid zugeordnete Katalogkennung;
Gehört die Katalogkennung zu einem Remote-System, zu dem eine RFA-Verbindung besteht, so wird der FILE-Aufruf mit der Parameterliste über RFA an das Remote-System geschickt.

userid
Benutzerkennung;
Default-Userid: die Benutzerkennung des Kommandos SET-LOGON-PARAMETERS bzw. LOGON.
Falls die Datei nicht bereits als mehrbenutzbar katalogisiert ist oder falls Speicherplatz freigegeben wird, darf eine fremde Benutzerkennung nur dann angegeben werden, wenn die aufrufende Task das Privileg TSOS besitzt oder Miteigentümer der Datei ist.

dateiname
vollqualifizierter Name einer Datei oder Dateigeneration. Für Banddateien kann er in Klammern am Ende eine Versionsnummer enthalten.

***DUMMY**
Der FILE-Makroaufruf beschreibt eine Pseudodatei (= DUMMY-Datei). Wenn gleichzeitig der LINK-Operand angegeben wird, wird ein TFT-Eintrag mit Datenträgerliste erstellt. In Zusammenhang mit dem TSET-Operanden wird auch ein TST-Eintrag erstellt. Alle übrigen Operanden im FILE-Makroaufruf werden nur auf formale Richtigkeit überprüft, aber nicht ausgewertet; es werden weder Geräte noch Datenträger angefordert oder Speicherplatz zugewiesen noch wird ein Katalogeintrag erstellt.

DUMMY-Datei als Eingabedatei: beim Leseversuch wird die Dateiende-Bearbeitung (EOF-Verarbeitung) angestoßen. DUMMY-Datei als Ausgabedatei: die Daten werden zwar in die Pufferbereiche des Programms übertragen, die Ausgabe auf einen Datenträger wird jedoch unterdrückt.

AVAIL = HIGH
Nur ab VERSION=3 und nur relevant für Dateien auf Pubsets und Net-Storage-Volumes:
Die Datei soll eine erhöhte Ausfallsicherheit haben und wird auf einem entsprechenden Volume-Set (z.B. DRV) angelegt.

Die Angabe wird in folgenden Fällen abgewiesen:
– die Datei belegt bereits Speicherplatz
– SPACE ist mit nichtpositiver Primärzuweisung angegeben
– WORKFIL=YES ist angegeben
– eine temporäre Datei ist angegeben
– die Datei kommt auf einem SF-Pubset ohne erhöhte Ausfallsicherheit zu liegen

– die Datei kommt auf einem SM-Pubset zu liegen, der kein Volume-Set mit erhöhter Ausfallsicherheit enthält
– die Datei kommt auf einer Privatplatte zu liegen
– es ist eine Banddatei angegeben

BLIM = zahl

Zum Neuerstellen von Banddateien mit Standardkennsätzen, die mit der Zugriffsmethode SAM verarbeitet werden sollen:
„zahl“ gibt an, wie viele Datenblöcke auf ein Band geschrieben werden dürfen.
$1 \leq$ zahl ≤ 999999.

Bei Erreichen des Grenzwerts wird Bandwechsel veranlasst (EOV-Verarbeitung); falls mit CHKPT-Operand gefordert, wird zuvor noch ein Fixpunkt an das Bandende geschrieben. Ist das Bandende erreicht, bevor die mit BLIM festgelegte Anzahl Blöcke geschrieben wurde, erhält der Benutzer eine Fehlermeldung im FCB.

Ist BLIM angegeben, so werden folgende Angaben abgewiesen: FCBTYPE=PAM/BTAM/ISAM, LABEL=NO/STD, FSEQ=n mit n>1 und FSEQ=UNK/NEW.

BLKCTRL

Nur ab VERSION=1:
legt fest, ob eine Datei des K-Formats (mit PAM-Schlüssel) oder des NK-Formats (ohne PAM-Schlüssel) zu verarbeiten ist. BLKCTRL ist relevant für das vorläufige Dateiformat.

Für die Verarbeitung von NK-SAM- und NK-PAM-Dateien stehen die gleichen Funktionen mit identischen Anwenderschnittstellen zur Verfügung wie für die entsprechenden K-Dateien. Bei NK-ISAM-Dateien bietet die Zugriffsmethode NK-ISAM Funktionen an, die über die des K-ISAM hinausgehen, z.B. die Verarbeitung von ISAM-Dateien in ISAM-Pools oder die Verwendung von Sekundärschlüsseln (siehe Handbuch „Einführung in das DVS“ [1]). Intern unterscheiden sich NK-ISAM- und K-ISAM-Verarbeitung, an der Anwenderschnittstelle ergeben sich jedoch nur geringfügige Änderungen bzgl. der Auswirkung ISAM-spezifischer Operanden (siehe nachfolgende Tabelle).

Operand	**BLKCTRL= PAMKEY**	**BLKCTRL= DATA/DATA2K/DATA4K**
DDEVICE DVOLUME DSPACE	Index-/Datentrennung bei ISAM-Dateien auf Privatplatten	Index-/Datentrennung wird nicht unterstützt, Operanden können jedoch angegeben werden
DUPEKY		Sätze mit gleichen Schlüsseln erhalten intern einen Zeitstempel
LOGLEN	Länge der logischen Markierung	Operand wird ignoriert
POOLLNK		Verbindung zu Benutzer ISAM-Pool (sonst: Standard-ISAM-Pool)

Operand	BLKCTRL= PAMKEY	BLKCTRL= DATA/DATA2K/DATA4K
OVERLAP	Leseoperationen werden überlappend durchgeführt (mit IOAREA2)	benachbarte Blöcke werden ebenfalls in den ISAM-Pool eingelesen
PAD	Mindestangabe	Maximal-Angabe für freien Bereich im Datenblock
SHARUPD	Blocksperren	Satz- oder Bereichssperren
VALLEN	Länge der Wertmarkierung	Operand wird ignoriert
VALPROP	Wertmarkierung wird ausgewertet	wird ignoriert

Wird der Operand BLKCTRL nicht angegeben (weder TFT noch FCB), so wird der Datei nach dem Eröffnen folgender BLKCTRL-Wert zugewiesen (sofern nicht der Wert aus dem Katalogeintrag übernommen wird):

BLKCTRL = PAMKEY für Dateien auf herkömmlichen (CKD-) Platten
BLKCTRL = DATA für SAM- oder ISAM-Dateien auf den neuen (FBA-) Platten (ohne PAM-Schlüssel-Simulation)
BLKCTRL = NO für PAM-Dateien auf den neuen (FBA-) Platten (ohne PAM-Schlüssel-Simulation)

= *BY-PROG

Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der BLKCTRL-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

= PAMKEY

Die Datei hat K-Format: Die Blockkontrollinformation wird außerhalb des Datenblocks in einem PAM-Schlüssel abgelegt. Eine solche K-Datei kann nicht auf einer NK-Platte (FBA-Platte ohne PAM-Schlüssel-Simulation) oder auf einem Net-Storage-Volume angelegt werden.

= NO

Die Angabe ist nur für PAM-Dateien und SAM-Banddateien sinnvoll; bei SAM-Plattendateien wird sie in BLKCTRL=DATA, bei ISAM-Dateien in BLKCTRL=DATA2K bzw. BLKCTRL=DATA4K umgewandelt.
Wird FCBTYPE=PAM angegeben, so wird eine NK-PAM-Datei angelegt, die keine blockspezifische Verwaltungsinformation enthält.
Diese Datei kann unabhängig von der gewählten logischen Blocklänge (BLKSIZE) sowohl auf K-Platte als auch auf NK2-Platte angelegt werden.
Wird als logische Blocklänge (BLKSIZE) ein Vielfaches von 4K angegeben (Blockungsfaktor „n“ geradzahlig), kann die Datei auch auf einer NK4-Platte angelegt werden.

= DATA

Die Datei hat NK-Format: Die Blockkontrollinformation steht zu Beginn eines jeden logischen Blockes (bei ISAM-Dateien am Beginn eines jeden 2-KByte- bzw. 4-KByte-Blocks). Eine NK-Datei kann sowohl auf K-Platte, auf NK2-Platte und bei entsprechend gewählter Blocklänge auch auf NK4-Platte liegen. Beim Neuanlegen einer Datei

(OPEN OUTPUT/OUTIN) wird eine NK2- oder eine NK4-Datei erstellt:
Für Dateien, die mit anderen Zugriffsmethoden als ISAM erstellt wurden, gilt in Abhängigkeit des Blockungsfaktors „n“ bei der Angabe der logischen Blocklänge mit BLKSIZE Folgendes:
– Ist der Blockungsfaktor n eine ungerade Zahl wird eine NK2-Datei angelegt
– Ist der Blockungsfaktor n eine gerade Zahl wird eine NK4-Datei angelegt

Beim Erstellen einer NK-ISAM-Datei (OPEN OUTPUT/OUTIN) wird das Dateiformat in Abhängigkeit des Plattenformats gewählt. Eine bereits geöffnete Datei kann unabhängig vom Blockformat geöffnet werden.

= DATA2K
Nur ab VERSION=2, für ISAM-Dateien:
Spezialisierung von „DATA“ für NK-ISAM-Dateien.

Beim Erstellen (OPEN OUTPUT/OUTIN) wird explizit eine NK2-ISAM-Datei erzeugt. Beim Eröffnen bestehender Dateien wird geprüft, ob es sich um eine NK2-ISAM-Datei handelt.

Die blockspezifische Verwaltungsinformation wird in den ersten 16 Byte eines jeden Datenblockes hinterlegt. Eine Datei mit dieser Angabe kann nicht auf einer NK4-Platte angelegt werden. Eine Datei, die auf NK4-Platte liegt, kann mit dieser Angabe nicht eröffnet werden.

= DATA4K
Nur ab VERSION=2, für ISAM-Dateien:
Spezialisierung von „DATA“ für NK-ISAM-Dateien.

Beim Erstellen (OPEN OUTPUT/OUTIN) wird explizit eine NK4-ISAM-Datei erzeugt. Beim Eröffnen bestehender Dateien wird geprüft, ob es sich um eine NK4-ISAM-Datei handelt.
Die blockspezifische Verwaltungsinformation wird in den ersten 16 Byte eines jeden 4-KByte-Blockes hinterlegt. Wird ein Blockungsfaktor n angegeben, so muss n eine gerade Zahl sein, d.h. die logische Blockgröße muss ein Vielfaches von 4-KByte betragen. Die Datei kann auf K-, NK2- und NK4-Platte angelegt bzw. dort eröffnet werden.

BLKSIZE
legt die Länge des logischen Blocks fest, d.h. die Länge der Übertragungseinheit von und zu den Ein-/Ausgabegeräten. BLKSIZE ist relevant für das vorläufige Dateiformat.

Wird der Operand BLKSIZE nicht angegeben (weder TFT noch FCB), so wird der Datei nach dem Eröffnen folgender BLKSIZE-Wert zugewiesen (sofern nicht der Wert aus dem Katalogeintrag übernommen wird):

K-/NK2-Datenträger	– BLKSIZE = STD
NK4-Datenträger	– BLKSIZE = (STD,2)

Für Plattendateien ergeben sich Wechselwirkungen mit dem SPACE- und dem RECSIZE-Operanden, für Banddateien mit dem LABEL-Operanden (siehe die beiden Tabellen unter BLKSIZE=länge auf Seite 472).

K-Plattendateien/Banddateien mit Standardblöcken: Logische Blöcke können aus mehreren PAM-Seiten bestehen. Das System verknüpft die zu einer Übertragungseinheit zusammengefassten PAM-Seiten automatisch.

Banddateien mit Nichtstandardblöcken: das Blockformat entspricht nicht dem des PPAM; der logische Block ist definiert durch die Anzahl Bytes, die pro Schreib-/Leseoperation geschrieben/gelesen werden.

= *BY-PROG
Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der BLKSIZE-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

= STD
entspricht der Angabe (STD,1); siehe unten.
Die Daten werden in Einheiten von 2048 Byte von/zu den Geräten übertragen; die für Anwenderdaten nutzbare Länge der Übertragungseinheit ist abhängig von der BLKCTRL-Angabe (bzw. dem Plattentyp).

= (STD,n)
„STD“ ist ein Standardblock der Größe 2048 Byte; „n“ ist der Blockungsfaktor ($1 \leq n \leq 16$)

Jeder logische Block besteht aus n PAM-Blöcken (1 PAM-Block/1 PAM-Seite = 2048 Byte), d.h. die max. Länge des logischen Blocks ist 16 PAM-Seiten = 32768 Byte. Für NK-Dateien legt „n“ die Länge des logischen Blockes als Vielfaches von 2048 Byte fest: Die Länge eines solchen Blockes beträgt n * 2048 Byte.

Für NK4-Dateien muss „n“ eine gerade Zahl sein, d.h. die Länge des logischen Blocks ist ein Vielfaches von 4-KByte. Für eine NK4-ISAM-Datei muss zusätzlich für den Operanden BLKCTRL der Operandenwert DATA4K angegeben sein.

Für SAM-Dateien, SETL-Verarbeitung: in jedem logischen Block dürfen höchstens 255 Sätze stehen, da die Positionierungsinformation nur 1 Byte lang ist. Diese Einschränkung entfällt bei der Verwendung eines 31-Bit-fähigen FCB.

= länge
Für Banddateien:
gibt die maximale Blocklänge in Byte an und legt gleichzeitig fest, dass die Datei aus Nichtstandardblöcken besteht, es wird kein PAM-Schlüssel geführt.
Es sind zum einen die Operanden BUFOFF und RECFORM zu berücksichtigen, zum anderen FCBTYPE und CHAINIO.

Operand RECFORM	Auswirkung
RECFORM=F	„länge“ gibt die Blocklänge einschließlich Länge der Pufferverschiebung an (siehe Operand BUFOFF); alle Blöcke haben dieselbe Länge
RECFORM=V/U	„länge“ gibt die maximale Blocklänge einschließlich der Länge der Pufferverschiebung (siehe Operand BUFOFF) an, d.h. die Blocklänge ist (wie die Satzlänge) variabel; gilt RECFORM=V zusammen mit CODE=EBCDIC oder LABEL=(STD,n) mit n > 1, muss „länge“ < 10000 sein

Operand FCBTYPE	zulässige Angabe für „länge“
SAM / BTAM	$1 \leq n \leq 32768$
PAM	--------

BUFOFF

Für Banddateien mit BLKCTRL=DATA oder SAM-Banddateien ohne Standardblockung:
legt die Pufferverschiebung (Buffer Offset) fest, d.h. die Länge eines Feldes, das am Anfang eines jeden Datenblocks eingefügt wird.

Wird der Operand BUFOFF nicht angegeben (weder TFT noch FCB), so wird der Datei nach dem Eröffnen folgender BUFOFF-Wert zugewiesen (sofern nicht der Wert aus dem Katalogeintrag übernommen wird):

– für Banddateien mit BLKCTRL=DATA

  bei FCBTYPE=SAM: BUFOFF=16
  bei FCBTYPE=PAM: BUFOFF=12

– für SAM-Banddateien ohne Standardblockung

  bei RECFORM=V: BUFOFF=4
  bei RECFORM=F/U: BUFOFF=0

= *BY-PROG

Ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der BUFOFF-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

= L

Der BUFOFF-Wert wird dem HDR2-Kennsatz der Datei entnommen. Ist kein HDR2-Kennsatz vorhanden oder enthält das Feld „Pufferverschiebung“ Leerzeichen (X'4040'), so treten dieselben Werte in Kraft wie bei fehlender BUFOFF-Angabe (weder TFT noch FCB).

= länge
gibt die Länge der „Pufferverschiebung“ an.

Für SAM-Dateien mit RECFORM=V gilt: 0 ≤ länge ≤ 4; ist BUFOFF=4, enthält dieses Feld die aktuelle Blocklänge. Bei Dateien mit BLKCTRL=DATA enthält es das Blockkontrollfeld.

BYPASS
Für Eingabedateien auf Band:
Wenn für die Benutzerkennung die Berechtigung dazu im Benutzerkatalog eingetragen ist, kann der Benutzer auf Kennsatzprüfung verzichten und angeben, wie das Band positioniert werden soll. Das DVS prüft, ob das richtige Band montiert ist, die Anwenderroutinen zur Kennsatzbehandlung werden normal aktiviert. Die Positionierungsangabe wird nur ausgewertet, wenn kein OPEN-Exit definiert ist.

Neben der Kennsatzprüfung entfällt auch die Codeprüfung.
Der Anwender muss bei CODE=OWN eigene Code-Tabellen zur Verfügung stellen.

BYPASS ermöglicht die Verarbeitung von Bändern, die unter anderen Betriebssystemen (z.B. BS1000) erstellt wurden oder deren Aufbau und Kennsatzformate dem System nicht bekannt sind. Die BYPASS-Angabe gilt nur während der Dateiverarbeitung, sie wird nicht in den Katalog aufgenommen.

In Zusammenhang mit BYPASS werden FSEQ- und SECLEV-Operand nicht ausgewertet.

= LP
Es erfolgt keine Kennsatzbehandlung; Anfangskennsätze werden weder geprüft noch gelesen; die Bandposition wird nicht verändert.

= (LP,n)
Es erfolgt keine Kennsatzbehandlung, das Band wird beim Eröffnen der Datei auf die n-te Abschnittsmarke ab Bandanfang positioniert; 0 ≤ n ≤ 32767

(LP,0): Positionieren auf Bandanfang

= (LP,+n)
Es erfolgt keine Kennsatzbehandlung; das Band wird bei Dateieröffnung um n Abschnittsmarken ab der aktuellen Bandposition vorpositioniert; 0 ≤ n ≤ 127

(LP,+0): das Band wird nicht neu positioniert

= (LP,-n)
Es erfolgt keine Kennsatzbehandlung; das Band wird bei Dateieröffnung um n Abschnittsmarken ab der aktuellen Bandposition zurückpositioniert;
0 ≤ n ≤ 127

(LP,-0): das Band wird nicht neu positioniert

CHAINIO = zahl
Für BTAM-Dateien bei geketteter Ein-/Ausgabe:
1 ≤ zahl ≤ 16
„zahl“ ist der Kettungsfaktor, der die Länge der Transporteinheit/Übertragungseinheit bei der Ein-/Ausgabe festlegt. „zahl“ bezeichnet dabei eine Anzahl Blöcke, sodass sich die Länge der Transporteinheit berechnet aus zahl * BLKSIZE.

Obwohl bei der Verarbeitung von BTAM-Dateien Angaben im Programm (BTAM-Makroaufruf) gegenüber dem Produkt „zahl“ * BLKSIZE überwiegen, muss CHAINIO im FILE-Kommando angegeben werden, wenn mit geketteter Ein-/Ausgabe gearbeitet wird.

CHKPT
Für Banddateien:
steuert, ob und wann automatisch ein Fixpunkt an das Bandende zu schreiben ist oder wie die Datei bei Wiederanlauf (RESTART-Kommando) weiterverarbeitet werden soll.

Voreinstellung: CHKPT=(NO,ACTIVE)

= (NO,...)
Es erfolgt keine automatische Fixpunktschreibung, sofern im FCB des Programms nichts anderes festgelegt ist.

= (BLIM,...)
Wenn das mit dem BLIM-Operanden gesetzte Blocklimit erreicht ist, wird automatisch ein Fixpunkt geschrieben; der Operand BLIM muss angegeben werden.

= (FEOV,...)
Bei jedem FEOV-Makroaufruf wird automatisch ein Fixpunkt geschrieben.

= (ANY,...)
Ein Fixpunkt wird automatisch geschrieben, wenn die mit BLIM gesetzte Grenze erreicht ist oder ein FEOV-Makro aufgerufen wird; der Operand BLIM muss angegeben werden.

= (...,DUMMY)
„pfadname“ wird bei einem Wiederanlauf mit dem Kommando RESTART-PROGRAM wie eine DUMMY-Datei behandelt.

= (...,ACTIVE)
Die Datei „pfadname“ wird bei einem Wiederanlauf (Kommando RESTART-PROGRAM) weiterverarbeitet.

CLOSE
Nur ab VERSION=2:
gibt an, mit welchem CLOSE-Modus die Datei geschlossen werden soll. Diese Angabe kann bei Dateischließung durch den CLOSE-Makroaufruf überschrieben werden.

Voreinstellung: Der CLOSE-Wert wird dem CLOSE-Makroaufruf entnommen.

Zur CLOSE-Verarbeitung siehe Handbuch „Einführung in das DVS“ [1].

= RWD
Für Bandverarbeitung:
positioniert das Band auf Bandanfang.

= REPOS
Für Bandverarbeitung:
positioniert das Band abhängig von der LABEL-Angabe auf den Anfang des aktuellen Dateiabschnitts.

= DISCON
Für Bandverarbeitung:
Das Band wird auf Bandanfang positioniert und entladen/freigegeben.

= LEAVE
Für Bandverarbeitung:
positioniert das Band abhängig von der LABEL-Angabe auf das logische Dateiende.

= INVAL
Die im Cache stehenden Blöcke der Datei werden invalidiert, d.h. als ungültig gekennzeichnet. Sie werden nicht auf die Platte zurückgeschrieben (Achtung bei Shared-Update-Verarbeitung).

= KEEP-DATA-IN-CACHE
Nur ab VERSION=3:
Im Cache stehende Blöcke der Datei werden nicht auf die Platte zurückgeschrieben, bleiben aber als gültig gekennzeichnet.

CLOSMSG
Nur ab VERSION=1:
Für sequenziell zu verarbeitende Dateien (SAM) kann der Anwender bestimmen, ob nach der CLOSE-Verarbeitung eine Abschlussmeldung ausgegeben werden soll (nach SYSOUT). Wird der Operand CLOSMSG nicht angegeben, so wird der Datei nach dem Eröffnen folgender CLOSMSG-Wert zugewiesen:

Platte: CLOSMSG = NO
Band: CLOSMSG = YES

= NO
Die Abschlussmeldung wird unterdrückt.

= YES
Die Abschlussmeldung wird ausgegeben.

CODE
Für Bandverarbeitung:
legt für SAM- oder BTAM-Dateien fest, ob und welche Umsetzungstabellen bei Ein-/Ausgabe verwendet werden.

Wird der Operand CODE nicht angegeben (weder TFT noch FCB), so wird der Datei nach dem Eröffnen folgender CODE-Wert zugewiesen (sofern nicht der Wert aus dem Katalogeintrag übernommen wird):

CODE = EBCDIC

Bei CODE=EBCDIC und CODE=ISO7 haben deutscher und internationaler Zeichensatz die gleiche Verschlüsselung.

Bei CODE=ISO7/OWN und FCBTYPE=SAM ist Folgendes zu beachten:
– die Blocklänge muss mit BLKSIZE=länge definiert werden, damit kein PAM-Schlüssel geschrieben wird;
– bei Ausgaben im Locate-Mode ändert sich bei variablem Satzformat (RECFORM=V) der Inhalt des Satzlängenfeldes.

= *BY-PROG
Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der CODE-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

= EBCDIC
Bei der Verarbeitung ist keine Code-Umsetzung erforderlich.

= ISO7
Die Banddatei ist/wird mit dem ISO-7-Bit-Code geschrieben, d.h. bei der Ausgabe wird EBCDI-Code in ISO-7-Bit-Code umgesetzt, bei der Eingabe ISO-7-Bit-Code in EBCDI-Code. Dabei wird die internationale ISO-Tabelle verwendet.

= ISO7D
Nur ab VERSION=3:
Die Banddatei ist/wird mit dem ISO-7-Bit-Code geschrieben, d.h. bei der Ausgabe wird EBCDI-Code in ISO-7-Bit-Code umgesetzt, bei der Eingabe ISO-7-Bit-Code in EBCDI-Code. Dabei wird die deutsche ISO-Tabelle verwendet.

= OWN
Die Umsetzung erfolgt über vom Benutzer erstellte Tabellen, deren Adressen im FCB-Makroaufruf angegeben werden müssen (siehe Makro FCB, Operanden TRTADR, TRTADW, Seite 444). Gleichzeitig muss im LABEL-Operanden die Kennsatzverarbeitung ausgeschaltet werden (LABEL=NO) oder mit LABEL=NSTD die Kennsatzverarbeitung ins Benutzerprogramm verlagert werden.

DATATTR = (*FROM-FILE,<c-string: filename 1..54>)

Nur ab VERSION=3:

Aus dem Katalogeintrag der hier angegebenen Referenzdatei werden beim Erzeugen eines TFT-Eintrags folgende Werte in den TFT-Eintrag übernommen, wobei allerdings explizit angegebene Werte Vorrang haben:

BLKCTRL, BLKSIZE, BUFOFF, CODE, FCBTYPE, KEYLEN, KEYPOS, LABEL, LOGLEN, RECFORM, RECSIZE, VALLEN, VALPROP

Die Referenzdatei muss im selben Pubset katalogisiert sein, wie die Datei, auf die sich der FILE-Aufruf bezieht. Der Aufrufer muss die Berechtigung haben, den Katalogeintrag der Referenzdatei (mit FSTAT oder /SHOW-FILE-ATTRIBUTES) zu lesen.

Die im Katalogeintrag der Referenzdatei enthaltenen Werte für BLKCTRL und BLKSIZE werden bei der Bildung des vorläufigen Dateiformats berücksichtigt.

Ist bei einem der oben aufgeführten Operanden der Wert *BY-PROG angegeben, so wird die Übernahme des entsprechenden Wertes aus dem Katalogeintrag der Referenzdatei unterdrückt und kein Wert in den TFT-Eintrag aufgenommen.

Beispiel

Es wird eine Referenzdatei angegeben, deren Katalogeintrag für BLKCTRL den Wert PAMKEY enthält. Dann gilt:

im FILE-Aufruf angegebener BLKCTRL-Wert	BLKCTRL-Wert im TFT-Eintrag
keiner	PAMKEY
*BY-PROG	keiner
DATA	DATA

DDEVICE = <name 1..8>

Für ISAM-Dateien mit Index-/Datentrennung:

Mit DDEVICE wird der Plattentyp für den Datenteil benannt (für den Indexteil mit DEVICE); mögliche Angaben für „gerät“ sind der Gerätetabelle im Handbuch „Systeminstallation [16] zu entnehmen. Die ab BS2000 V9.5 neu eingeführten Gerätetypen werden nur ab VERSION=1 unterstützt. DDEVICE muss angegeben werden, wenn für die Datei noch kein Speicherplatz reserviert wurde. Ist DDEVICE angegeben, so müssen auch DVOLUME und DSPACE angegeben werden.

Ist bei DVOLUME mindestens ein Datenträgerkennzeichen angegeben, wird jede Angabe eines dem System bekannten Plattengerätetyps wie die Angabe STDDISK behandelt.

NK-ISAM unterstützt keine Index-/Datentrennung, DDEVICE kann aber angegeben werden (Kompatibilität zu K-ISAM).

DESTOC

Für Bandverarbeitung ab VERSION=1:

Der Anwender kann bestimmen, ob im Anschluss an die EOF-/EOV-Verarbeitung weitere auf dem Band stehende Daten durch Überschreiben gelöscht werden sollen.

DESTOC wirkt sich nur aus, wenn für „pfadname“ mit dem LINK-Operanden ein TFT-Eintrag eingerichtet wird.

Wird der Operand DESTOC nicht angegeben, so wird beim Eröffnen der Datei die DESTROY-Angabe aus dem Katalogeintrag übernommen.

DESTOC hat die gleiche Funktion wie der Operand DESTROY im CATAL-Makro, die DESTOC-Angabe hat jedoch Vorrang vor dem DESTROY-Wert im Katalogeintrag. Der Wert für DESTOC wird nicht in den Katalogeintrag übernommen.

= NO
Das Löschen bis zum Bandende unterbleibt.

= YES
Nach Schreiben der EOF-/EOV-Kennsätze wird der Rest der Daten bis zum Bandende gelöscht.

DEVICE

definiert den Plattengerätetyp bzw. Bandtyp.
Wenn bei VOLUME eine Privatplatte angegeben wird, die nicht im MAREN-Katalog steht, dann muss der Operand DEVICE angegeben werden.

Voreinstellungen

Wenn die Datei vor dem FILE-Aufruf noch keinen Speicherplatz hat und durch den FILE-Aufruf Speicherplatz erhält, dann gilt:

– Ist DEVICE=STDDISK angegeben und weder VOLUME noch NFTYPE angegeben, so wird die Datei auf gemeinschaftlichen Platten angelegt (und auch bei künftigen Erweiterungen nicht auf einem Net-Storage-Volume):

– Ist DEVICE=NETSTOR (der Volumetyp für Net-Storage-Volumes) angegeben und VOLUME nicht angegeben, so wird die Datei auf einem Net-Storage-Volume angelegt.

Wenn weder DEVICE noch VOLUME angegeben sind, gilt:
– Ohne Angabe von NFTYPE wird die Datei auf gemeinschaftlicher Platte angelegt.
– Bei Angabe von NFTYPE wird die Datei des angegebenen Dateityps auf einem beliebigen Net-Storage-Volume angelegt, sofern ein solches vorhanden ist.

Künftige Erweiterungen können hier ein abweichendes Verhalten ermöglichen.

= <name 1..8>
bestimmt den Gerätetyp bei Plattengeräten bzw. den Volumetyp bei Net-Storage-Volumes und Bandgeräten.

Mögliche Angaben für Plattengeräte sind der Gerätetabelle im Handbuch „Systeminstallation [16] zu entnehmen (Spalte Gerätetyp), zulässige Werte für Bandgeräte der Volumetyp-Tabelle (siehe Handbuch „Kommandos [3]).

DEVICE=NETSTOR (der Volumetyp für Net-Storage-Volumes) spezifiziert ein Net-Storage-Volume.

Mit DEVICE=TAPE können keine Magnetbandkassetten angefordert werden.

Wird beim Einrichten einer Datei für DEVICE ein Bandtyp angegeben, jedoch für VOLUME keine Angabe gemacht, so wird bei der OPEN-Verarbeitung ein freies Band mit Standardkennsätzen (SCRATCH-TAPE) angefordert und vom Operator zugewiesen. Ein Band ist aus der Sicht des DVS frei, wenn es entweder noch nicht beschrieben wurde oder wenn die Sperrfrist der ersten Datei auf dem Band abgelaufen und Schreibzugriff erlaubt ist.

Ist bei VOLUME mindestens ein Datenträgerkennzeichen angegeben, wird jede Angabe eines dem System bekannten Plattengerätetyps wie die Angabe STDDISK behandelt.

= WORK
Nur für Bandverarbeitung:
bewirkt, dass bei der OPEN-Verarbeitung ein Arbeitsband mit Standardkennsätzen angefordert wird.
Arbeitsbänder sind keinem Eigentümer zugeordnet, das entsprechende Feld im VOL1-Kennsatz enthält stets Leerzeichen (X'40'). Arbeitsbänder sollten nur dann angefordert werden, wenn sie nur während der Verarbeitung benötigt werden und nicht archiviert werden sollen. Auf Arbeitsbändern ist kein Dateischutz möglich. Arbeitsbänder werden bei der Anforderung vom Operator zugewiesen, Angaben im VOLUME-Operanden werden ignoriert. Die Operanden TSET und STATE=FOREIGN dürfen nicht zusammen mit DEVICE=WORK angegeben werden.

Für Mehrbanddateien sollte DEVICE=WORK nicht angegeben werden, da immer automatisch das gerade zur Verfügung stehende Arbeitsband zugewiesen wird.

Magnetbandkassetten können nicht als Arbeitsband angefordert werden.

DISKWR
Nur ab VERSION=3 und nur relevant für Dateien auf Pubsets oder Net-Storage-Volumes:
gibt an, zu welchem Zeitpunkt nach einer Schreiboperation sich die Daten der Datei in einem konsistenten Zustand befinden müssen. Für Dateien auf Pubsets oder Net-Storage-Volumes wird die Angabe in den Katalogeintrag übernommen. Für Dateien auf SM-Pubsets wird die Angabe bei der Auswahl des Volume-Sets berücksichtigt.

Falls DISKWR nicht angegeben ist und die Datei noch keinen Katalogeintrag hat, wird für eine permanente Datei der Wert IMMEDIATE angenommen und für eine temporäre Datei der Wert BY-CLOSE. Falls DISKWR nicht angegeben ist und die Datei bereits einen Katalogeintrag hat, wird der im Katalogeintrag enthaltene Wert für DISKWR verwendet.

Die Angabe wird in folgenden Fällen abgewiesen:
– die Datei belegt bereits Speicherplatz
– SPACE ist mit nichtpositiver Primärzuweisung angegeben

= IMMEDIATE
Die Daten der Datei müssen sich unmittelbar nach Beendigung einer Schreiboperation in konsistentem Zustand befinden.

= BY-CLOSE
Die Daten der Datei müssen sich erst nach Dateischließung in konsistentem Zustand befinden. Damit kann die Datei über einen flüchtigen Schreib-Cache bearbeitet werden.

DSPACE
In Zusammenhang mit DDEVICE/DVOLUME für den Datenteil von ISAM-Dateien mit Index-/Datentrennung:
DSPACE legt Speicherplatzzuweisungen für den Datenteil einer ISAM-Datei fest. Die Regeln für die Angabe von Primär- und Sekundär- sowie Absolutzuweisung entsprechen denen des Operanden SPACE, beziehen sich jedoch auf im Operanden DVOLUME genannte Datenträger (siehe auch Operanden DDEVICE, DVOLUME sowie Abschnitt „Index-/Datentrennung“, Handbuch „Einführung in das DVS“ [1]). NK-ISAM unterstützt keine Index-/Datentrennung; DSPACE kann jedoch angegeben werden (Kompatibilität zu K-ISAM).

= <integer 0..2147483647>
Primärzuweisung, sofort wirksam

= (<integer 0..2147483647>,<integer 0..32767>)
Primärzuweisung wird sofort wirksam, der Wert der Sekundärzuweisung wird in den Katalogeintrag übernommen; 0 ≤ sekundär ≤ 32767

= (<integer 0..2147483647>,<integer 0..2147483647>,ABS)
ABS: Absolutzuweisung;
angegeben werden die PAM-Seitennummer, an der die Absolutzuweisung beginnt, gefolgt von der Anzahl PAM-Seiten, die reserviert werden sollen.

DUPEKY
Für ISAM-Dateien:
legt fest, ob in einer Datei mehrere Sätze mit gleichen Primärschlüsselwerten auftreten dürfen.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

= NO
Identische Primärschlüsselwerte dürfen nicht in verschiedenen Sätzen der Datei auftreten.

= YES
Haben mehrere Sätze den gleichen Primärschlüsselwert, überschreiben sie sich nicht gegenseitig, sondern werden in der Reihenfolge ihrer Erstellung hintereinander geschrieben. Der Operand DUPEKY=YES ist nur von Bedeutung, wenn die ISAM-Datei mit PUT-Makroaufruf sequenziell erstellt oder mit STORE-Makroaufruf nichtsequenziell erweitert wird. Der INSRT-Makroaufruf kann nicht dazu verwendet werden, Sätze mit gleichen Primärschlüsselwerten zu schreiben.
Zu DUPEKY=YES siehe auch Seite 425.

DVOLUME
In Zusammenhang mit DDEVICE für K-ISAM-Dateien auf Privatplatte – mit Index-/Datentrennung:
DVOLUME gibt die Archivnummer („vsn“) des Datenträgers an, auf dem der Datenteil der ISAM-Datei gespeichert werden soll; für den Indexteil ist der Operand VOLUME anzugeben. Es gelten analog die Erläuterungen wie bei DDEVICE. NK-ISAM unterstützt keine Index-/Datentrennung, DVOLUME kann jedoch angegeben werden (Kompatibilität zu K-ISAM). Für Dateien auf einem Net-Storage-Volume ist Index-Daten-Trennung nicht möglich.

= <@adr>
Nur ab VERSION=2:
adr ist eine symbolische Adresse im Programm, an der mit dem Makroaufruf FILELST DVOLUME=... eine DVOLUME-Liste angelegt wurde.
Das Zeichen „@“ ist Bestandteil des Operandenwertes und muss mit angegeben werden.

= PRIVATE
fordert an der Konsole die Bereitstellung einer Privatplatte an.

= (PRIVATE,<integer 1..9>)
fordert an der Konsole die Bereitstellung der gewünschten Anzahl Privatplatten an.

= list-poss(255): <name 1..6>
Für den Datenteil der ISAM-Datei werden die mit ihrer Archivnummer angegebenen Privatplatten benötigt.

EXC32GB
Nur für Plattendateien; nur für Nicht-Pamkey-Dateien:
Der Operand EXC32GB bestimmt, ob die Dateigröße bei der Datenverarbeitung 32 GB überschreiten darf oder nicht (siehe Seite 109). Der Operand wird in die TFT (Task File Table) eingetragen und erst beim Öffnen der Datei mit OPEN ausgewertet.
EXC32GB hat keinen Einfluss auf die Speicherplatzzuweisung beim FILE-Aufruf.

= FORBIDDEN
Die Dateigröße darf 32 GB nicht überschreiten.

= ALLOWED
Die Dateigröße darf 32 GB überschreiten.

FCBTYPE
bestimmt die Zugriffsmethode bei der Dateiverarbeitung.

= *BY-PROG
Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der FCBTYPE-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

= ISAM
„pfadname“ ist eine ISAM-Datei. In Abhängigkeit vom Operanden BLKCTRL wird sie als NK-ISAM-Datei (BLKCTRL=DATA) oder als K-ISAM-Datei (BLKCTRL=PAMKEY) verarbeitet. Die Zugriffsmethode ISAM ist im Handbuch „Einführung in das DVS“ [1] beschrieben.

ISAM-spezifische Operanden: DUPEKY, KEYLEN, KEYPOS, LOGLEN, POOLLNK, VALLEN, WROUT sowie DDEVICE, DSPACE, DVOLUME und VALPROP.

= BTAM
„pfadname“ ist eine Banddatei, die mit der Zugriffsmethode BTAM verarbeitet werden soll. Die Zugriffsmethode BTAM ist im Handbuch „Einführung in das DVS“ [1] beschrieben.

BTAM-spezifische Operanden: CHAINIO, OPEN=SINOUT, STREAM

= PAM
„pfadname“ ist eine PAM-Datei, sie wird mit der Zugriffsmethode UPAM verarbeitet. Die Zugriffsmethode UPAM ist im Handbuch „Einführung in das DVS“ [1] beschrieben.

PAM-Dateien können auf Band oder auf Platte gespeichert sein.

= SAM
„pfadname“ ist eine SAM-Datei auf Platte oder Band. SAM-Dateien werden in der Regel sequenziell verarbeitet mit den Zugriffsmethoden SAM oder auch UPAM. Die Zugriffsmethode SAM ist im Handbuch „Einführung in das DVS“ [1] beschrieben.

SAM-spezifische Operanden: CLOSMSG, OPEN=UPDATE

FSEQ

Für Banddateien, die zu einer Dateimenge (File Set) gehören:
gibt die (laufende) Nummer einer Datei innerhalb der Dateimenge an. Sind z.B. auf einem Band mehrere Dateien gleichen Namens gespeichert, wird der Zugriff über FSEQ gesteuert. Dies gilt auch für MF/MV-Sets.

Liegt bei der Dateieröffnung weder im TFT-Eintrag noch im FCB ein FSEQ-Wert vor, so wird bei bisher nicht eröffneten Dateien im Katalogeintrag FSEQ=1 eingetragen. Bei Dateien, die schon früher eröffnet wurden, wird der FSEQ-Wert aus dem Katalogeintrag übernommen.

Ist FSEQ als Null-Operand angegeben, so muss die Datei bereits einen Katalogeintrag haben. Ist dort eine Dateifolgenummer eingetragen, so wird diese in den TFT-Eintrag übernommen. Andernfalls wird im Katalogeintrag keine Dateifolgenummer und im TFT-Eintrag der Null-Operand eingetragen.

= UNK
Falls die Datei bereits einen Katalogeintrag hat und dort eine Dateifolgenummer eingetragen ist, wird diese in den TFT-Eintrag übernommen. Andernfalls wird im Katalogeintrag keine Dateifolgenummer und im TFT-Eintrag UNK eingetragen. Für eine Foreign-Banddatei mit Standardkennsätzen bedeutet dies bei der Dateieröffnung, dass das Band nach der Datei durchsucht und entsprechend positioniert wird.

= NEW
Nur zulässig falls noch kein Erstellungsdatum im Katalogeintrag eingetragen ist:
Der Wert NEW wird in den TFT-Eintrag, nicht jedoch in den Katalogeintrag eingetragen. Falls im Katalogeintrag bereits eine Dateifolgenummer eingetragen ist, wird sie gelöscht. Für eine noch nicht existierende Banddatei mit Standardkennsätzen bedeutet NEW bei der Dateieröffnung, dass die Datei hinter das bisherige Ende der Dateimenge geschrieben und die „Dateifolgenummer“ um 1 erhöht wird.

= <integer 0..9999>
Falls die Datei einen Katalogeintrag mit Erstellungsdatum hat, muss die FSEQ-Angabe mit der Dateifolgenummer im Katalogeintrag übereinstimmen. Andernfalls (insbesondere für Foreign-Dateien) wird die Zahl im Katalogeintrag und im TFT-Eintrag als Dateifolgenummer eingetragen. Bei der Dateieröffnung wird das Band entsprechend der Dateifolgenummer positioniert.

FSEQ=0 bezeichnet wie FSEQ=1 die erste Datei der Dateimenge.

IOPERF

Nur ab VERSION=2 und nur relevant für Dateien/Dateigenerationen auf gemeinschaftlichen Datenträgern oder Net-Storage-Volumes:

Gibt das Performance-Attribut der Datei an. Es bestimmt welche Priorität für die im Operanden IOUSAGE bezeichneten Ein-/Ausgabe-Operationen gewünscht wird. Das höchste zulässige Performance-Attribut ist im Benutzereintrag festgelegt (siehe Ausgabe des Kommandos SHOW-USER-ATTRIBUTES).

Beim Katalogisieren der Datei wird die Angabe mit dem höchsten zulässigen Performance-Attribut abgeglichen und in den Katalogeintrag übernommen bzw. STD eingetragen, falls IOPERF nicht angegeben ist.

Für eine katalogisierte Datei werden die entsprechenden Angaben im Katalogeintrag nicht verändert. Ist beim LINK-Operanden ein Dateikettungsname angegeben, wird, falls IOPERF angegeben wurde, der Wert in den TFT-Eintrag übernommen.

Beim Neuanlegen einer Datei auf einem SM-Pubset wird die Performance-Eigenschaft bei der Selektion des Volume-Sets berücksichtigt (z.B. Selektion eines Volume-Sets, dem ein Cache zugeordnet ist).

= STD
Die Datei wird nicht über einen Cache bearbeitet.

= VERY-HIGH
Die Datei wird, wenn möglich, über einen Cache bearbeitet. Wenn möglich, wird die gesamte Datei permanent in einem Cache gehalten (höchste Performance-Priorität).

= HIGH
Die Datei wird, wenn möglich, über einen Cache bearbeitet.

= USER-MAX
Die Datei wird entsprechend dem höchsten Performance-Attribut verarbeitet, das für die Benutzerkennung im Benutzerkatalog eingetragen ist.

IOUSAGE

Nur ab VERSION=2 und nur relevant für Dateien/Dateigenerationen auf gemeinschaftlichen Datenträgern oder Net-Storage-Volumes:

gibt an, auf welche I/O-Operationen sich das Performance-Attribut (IOPERF) der Datei bezieht. Beim Katalogisieren der Datei wird die Angabe in den Katalogeintrag übernommen bzw. RDWRT eingetragen, falls IOUSAGE nicht angegeben ist.

Für eine katalogisierte Datei werden die entsprechenden Angaben im Katalogeintrag nicht verändert.

Ist beim LINK-Operanden ein Dateikettungsname angegeben, wird, falls IOUSAGE angegeben wurde, der Wert in den TFT-Eintrag übernommen.

Beim Neuanlegen einer Datei auf einem SM-Pubset wird der IOUSAGE-Wert bei der Selektion des Volume-Sets berücksichtigt (z.B. Selektion eines Volume-Sets, dem ein Lese-Cache zugeordnet ist).

= RDWRT
Das Performance-Attribut bezieht sich sowohl auf Lese- wie auf Schreiboperationen.

= WRITE
Das Performance-Attribut bezieht sich nur auf Schreiboperationen.

= READ
Das Performance-Attribut bezieht sich nur auf Leseoperationen.

KEYLEN = länge
Für ISAM-Dateien:
gibt die Länge des ISAM-Schlüssels an.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

= *BY-PROG
Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der KEYLEN-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

= <integer 1..255>
Länge des ISAM-Schlüssels in Byte.

KEYPOS = zahl
Für ISAM-Dateien:
gibt die Position des Primärschlüssels im Datensatz an. Bei Sätzen variabler Länge müssen 4 Byte für Satzlängen- und Steuerfeld berücksichtigt werden. Der Primärschlüssel kann an beliebiger Stelle im Datensatz stehen, jedoch innerhalb einer Datei immer an derselben Position.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

= *BY-PROG
Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der KEYPOS-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

= <integer 1..255>
Byte-Position des Primärschlüssels.

LABEL
Für Banddateien:
legt die Kennsatzeigenschaften für Dateien auf Magnetband oder Magnetbandkassette fest; wie die Kennsätze verarbeitet werden, hängt vom SECLEV-Operanden ab.

Für bereits bestehende Banddateien gilt immer der im VOL1-Kennsatz angegebene Normvermerk; für Ausgabedateien (OPEN OUTIN/OUTPUT) wird der LABEL-Operand ausgewertet. Enthält das Band schon Dateien oder Dateiabschnitte, wird der Normvermerk im VOL1-Kennsatz entsprechend der LABEL-Angabe versorgt/geändert.

= *BY-PROG
Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der LABEL-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

= STD
Datei und Datenträger erhalten/haben Standardkennsätze, entsprechend DIN-Norm 66029, Austauschstufe 1.

= (STD,<integer 0..3>)
Datei und Datenträger erhalten/haben Standardkennsätze entsprechend der angegebenen Austauschstufe der DIN-Norm 66029; die Austauschstufe 4 ist in Vorbereitung. Zur Angabe und Auswirkung des Operanden siehe auch „Auswirkungen des Operanden LABEL“ auf Seite 432.

= NO
Dateikennsätze werden weder gelesen noch geschrieben (keine Dateikennsatzverarbeitung). Hat das Band Standardkennsätze, verarbeitet das System die Bandkennsätze und prüft die Zugriffsrechte.

= NSTD
Die Banddatei hat/erhält Nichtstandardkennsätze, die Dateikennsatzverarbeitung erfolgt im Benutzerprogramm. Hat der Datenträger Standardkennsätze, führt das System Bandkennsatzverarbeitung durch und prüft die Zugriffsrechte.

LINK = <name 1..8>
Für den hier angegebenen Dateikettungsnamen („name“) wird ein TFT-Eintrag angelegt, die übrigen Operanden werden ausgewertet und die Werte in den TFT-Eintrag übernommen (außer SPACE, DSPACE, AVAIL, WORKFIL, VOLSET, STOCLAS und DISKWR). Ggf. werden Datenträger aus der Datenträgerliste angefordert.

Existiert bereits ein Eintrag gleichen Namens in der TFT, wird er zunächst implizit freigegeben und anschließend mit den aktuellen Werten im FILE-Aufruf neu aufgebaut. Der „alte“ TFT-Eintrag darf nicht im Zustand „aktiv“ sein. Wurde der alte TFT-Eintrag mit einem LOCK-FILE-LINK-Kommando gesperrt, so bleibt auch der neue Eintrag gesperrt. Außerdem werden die alten Datenträger- und Gerätereservierungen aufgehoben; Bandgeräte bleiben jedoch dem Auftrag verfügbar. Über Dateikettungsname/TFT werden Programm und Datei miteinander verknüpft.

Der TFT-Eintrag wird in der Task des Aufrufers angelegt, sofern nicht der RFA-Fall-vorliegt. Falls die Katalogkennung im Pfadnamen zu einem Remote-System gehört, zu dem eine RFA-Verbindung besteht, wird der TFT-Eintrag in der Remote-Task angelegt, parallel dazu wird auch in der Task des Aufrufers ein TFT-Eintrag angelegt, wobei im TFT-Eintrag der Aufrufertask gegenüber dem TFT-Eintrag der Remote-Task Informationen fehlen können, z.B.:

– Benutzerkennung im Pfadnamen der Datei
– Informationen, die nicht über Operanden des FILE-Makros angebbar sind
– Informationen, die die Volume Table des TFT-Eintrags betreffen
– Angaben zu IOPERF, IOUSAGE, DEVICE, DDEVICE, FSEQ, MOUNT
– mit dem Operanden DATATTR vom Katalogeintrag der Referenzdatei übernommene Werte

Soll der Dateikettungsname über die Kommandoschnittstelle ansprechbar sein, muss er dem Datentyp <structured_name 1..8> entsprechen (siehe Handbuch „Kommandos“ [3]).

Hinweis

Ist der LINK-Operand nicht angegeben, wird kein TFT-Eintrag erstellt.
Die meisten Operanden des FILE-Aufrufs werden nur im Zusammenhang mit einem TFT-Eintrag ausgewertet. Eine Ausnahme bilden diejenigen Operanden, deren Werte in den Katalogeintrag übernommen werden oder die die FILE-Verarbeitung steuern, wie:
IOPERF, IOUSAGE, DEVICE, VOLUME, SPACE, DDEVICE, DVOLUME, DSPACE, FSEQ (nur teilweise) MOUNT und STATE=FOREIGN.

LOCKENV
Nur ab VERSION=3:
legt in Abhängigkeit vom Eröffnungsmodus und vom SHARUPD-Wert fest, ob die Datei bei der Verarbeitung gleichzeitig von verschiedenen Systemen aus zum Schreiben geöffnet werden kann.

= HOST
Die Datei kann bei der Verarbeitung nicht gleichzeitig von verschiedenen Systemen aus zu Schreiben geöffnet sein.

= XCS
Die Datei kann bei der Verarbeitung gleichzeitig von verschiedenenen Systemen aus mit SHARUPD=YES zum Schreiben geöffnet sein, wenn beide Systeme demselben XCS-Verbund angehören.

LOGLEN = <integer 0..255>
Für ISAM-Dateien:
bestimmt die Länge einer logischen Markierung im ISAM-Index in Byte; die maximale Länge ergibt sich aus der Länge des ISAM-Schlüssels und einer evtl. vorhandenen Wertmarkierung (siehe Operand VALLEN, Seite 506), da der gesamte ISAM-Index höchstens 255 Byte lang sein darf.

Es gilt also:

LOGLEN ≤ 255 – KEYLEN – VALLEN

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

Der ISAM-Index kann im Anschluss an den ISAM-Schlüssel eine logische Markierung (Logical Flag) enthalten, mit der Selektionseigenschaften bitweise dual verschlüsselt werden. In K-ISAM-Dateien werden alle logischen Markierungen eines Blocks ausgewertet und das Resultat in den nächsthöheren Indexeintrag übernommen. NK-ISAM unterstützt die Flagverarbeitung kompatibel, übernimmt die Flags jedoch nicht in den Indexeintrag. Die LOGLEN-Angabe wird ignoriert.

= *BY-PROG
Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der LOGLEN-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben. In allen Makroaufrufen, die sich durch den MF-Operanden unterscheiden (MF=L/E/D/C/S), muss der Versionsoperand den gleichen Wert haben. MF=S ist nur für VERSION=0 zulässig.

Voreinstellung: für VERSION=0 ist MF=S Voreinstellung.

MOUNT
gibt an, welche Datenträger aus der Datenträgerliste (siehe Operand VOLUME) von der FILE-Verarbeitung angefordert werden sollen.

Die Wechselwirkung mit dem Operanden VSEQ ist zu berücksichtigen:
– VSEQ: MOUNT-Angaben müssen größer oder gleich der VSEQ-Angabe sein.
  Ausnahme: MOUNT=0.
  Bei VSEQ=n muss die MOUNT-Liste mit „n“ beginnen (MOUNT=(n[,n+1][,n+2][,...);
  Bei VSEQ=(L=(n_1, n_2,...)) muss die MOUNT-Liste aus den ersten k Elementen der VSEQ-Liste bestehen (MOUNT=(n_1, n_2,...,n_k)).
– Falls der VSEQ-Operand nicht angegeben wurde, muss die MOUNT-Liste mit „1“ beginnen (MOUNT = 1, 2, ..., k)).
  Ausnahme: MOUNT=0.

Für die Anforderung von gemeinschaftlichen Platten gilt:
- Ist LINK nicht angegeben, so werden keine Platten angefordert.
- Ist die Datei migriert, so werden keine Platten angefordert.
- Ist MOUNT=0 angegeben, so werden keine Platten angefordert.
- Ist LINK angegeben, so wird jede MOUNT-Angabe außer MOUNT=0 abgewiesen.
- Ist LINK ohne MOUNT angegeben und ist die Datei nicht migriert, so werden alle Platten der Datenträgerliste angefordert.

Für die Anforderung von Privatplatten gilt:
- Wenn mindestens einer der Operanden SPACE, VOLUME, DSPACE oder DVOLUME angegeben ist sowie bei REUSE (Verwendung von Datenträgern der ältesten Generation beim Anlagen einer neuen Generation) wird die Angabe MOUNT=0 ignoriert.
- Falls MOUNT=0 wirksam wird, wird keine Privatplatte angefordert.
- Falls MOUNT=0 nicht wirksam wird und LINK nicht angegeben ist, werden die erste Privatplatte mit Extent und ggf. zusätzlich die für die Speicherplatzzuweisung benötigten Privatplatten angefordert.
- Ist LINK angegeben und keine MOUNT-Angabe wirksam, so werden alle Platten aus der Datenträgerliste angefordert.
- Ist LINK angegeben und sind in MOUNT k Zahlen ungleich 0 angegeben, so werden die ersten k Platten aus der Datenträgerliste angefordert.

Für die Anforderung von Net-Storage-Volumes gilt:
- Wenn MOUNT=0, aber weder SPACE noch VOLUME angegeben ist und die Datei schon vor dem FILE-Aufruf auf einem Net-Storage-Volume liegt, dann wird kein Net-Storage-Volume angefordert.
- In allen anderen Fällen wird das Net-Storage-Volume angefordert, auf dem die Datei liegt bzw. angelegt wird.

Für die Anforderung von Bändern gilt:
- Ist DEVICE=WORK, so werden keine Bänder angefordert.
- Ist MOUNT=0 oder weder LINK noch MOUNT angegeben, so werden keine Bänder angefordert.
- Ist MOUNT≠0 angegeben, so werden die Bänder gemäß MOUNT-Liste angefordert.
- Ist LINK ohne MOUNT angegeben, so wird genau ein Band aus der Datenträgerliste angefordert, und zwar bei Angabe von VSEQ=n das n-te Band, bei Angabe von VSEQ=(L=(n_1, n_2,...)) das n_1-te Band, bei fehlender VSEQ-Angabe das erste Band.
- Jede Zahl n>0 in der MOUNT-Liste bezieht sich auf das n-te Band der Datenträgerliste.

= 0

Für Plattendateien:
der Datenträger wird erst zum OPEN-Zeitpunkt angefordert, sofern weder VOLUME/DVOLUME noch SPACE/DSPACE angegeben wurden.
für Banddateien:
Das Band wird erst zum OPEN-Zeitpunkt angefordert.

= @adr
Nur ab VERSION=2:
adr ist eine symbolische Adresse im Programm, an der mit dem Makroaufruf FILELST MOUNT=... eine MOUNT-Liste angelegt wurde.
Das Zeichen „@“ ist Bestandteil des Operandenwertes und muss angegeben werden.

= list-poss(255): <integer 1..255>
Jede angegebene Zahl n bezieht sich auf den n-ten Datenträger der Datenträgerliste.

NFTYPE
Nur ab VERSION=3 und nur relevant für Dateien auf Net-Storage-Volumes:
gibt den Dateityp für die anzulegende Net-Storage-Datei an.
Wenn diese Angabe den Angaben im Operanden DEVICE und VOLUME widerspricht (z.B. Angabe einer Privatplatte), wird der Makroaufruf mit Fehler abgebrochen. Wenn die Operanden DEVICE und VOLUME nicht angegeben sind, wird die Datei mit dem angegebenen Dateityp auf einem beliebigen Net-Storage-Volume (falls vorhanden) angelegt.

= BS2000
Die Datei wird auf Net-Storage als BS2000-Datei angelegt.

= NODE-FILE
Die Datei wird auf Net-Storage als Node-File angelegt.

OPEN
gibt an, mit welchem OPEN-Modus die Datei eröffnet werden soll. Diese Angabe kann bei Dateieröffnung durch den OPEN-Makroaufruf überschrieben werden.
Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

Zu den zulässigen Angaben bei den jeweiligen Zugriffsmethoden siehe auch die Tabelle „OPEN-Modi beim Makro FCB“ auf Seite 434. Die einzelnen OPEN-Modi sind auch unter den entsprechenden Zugriffsmethoden detaillierter beschrieben.

= INPUT
„pfadname“ ist eine Eingabedatei, d.h. sie muss vorhanden sein.

= EXTEND
Eine vorhandene Datei wird erweitert, d.h. an das Dateiende werden weitere Datenblöcke angefügt, oder die Datei wird ab einem bestimmten Punkt überschrieben; es sind nur sequenzielle Schreiboperationen zulässig. Bei Banddateien werden abhängig von der LABEL-Angabe Kennsätze erzeugt.

= INOUT
Eine vorhandene Datei wird für nichtsequenzielle Verarbeitung eröffnet; es sind Schreib- und Leseoperationen zulässig. Bei Bandverarbeitung ist nach Abschluss der OPEN-Verarbeitung das Band auf Bandanfang positioniert; es werden keine Kennsätze geschrieben.

= OUTIN
Die Datei wird erstellt oder – falls bereits vorhanden – ab Dateianfang überschrieben. Es sind sowohl Schreib- als auch Leseoperationen zulässig (nichtsequenziell).
Für Banddateien werden Kennsätze geschrieben.

= OUTPUT
Die Datei wird sequenziell erstellt oder – falls bereits vorhanden – ab Dateianfang überschrieben. Für Banddateien werden Kennsätze geschrieben.

= REVERSE
Die Datei „pfadname“ muss bereits vorhanden sein; sie wird als Eingabedatei für sequenzielles Lesen mit Verarbeitungsrichtung Dateiende → Dateianfang eröffnet.
Bei Plattendateien darf sich die Datei nicht über mehrere Platten erstrecken.
Bei Banddateien ist kein automatischer Spulenwechsel möglich. Ein einzelner Dateiabschnitt kann verarbeitet werden (das entsprechende Band ist ggf. mit VSEQ auszuwählen). Banddateien sind nach Abschluss der OPEN-Verarbeitung auf das Ende des Dateiabschnitts positioniert.

= SINOUT
Nur für BTAM-Banddateien:
Die Datei muss vorhanden sein, das Band darf nicht auf Bandanfang positioniert sein; Datenblöcke können gelesen oder geschrieben werden. Im Gegensatz zu INOUT wird das Band nicht positioniert.

= UPDATE
Nur für SAM-Plattendateien:
Die Sätze der Datei können mittels GET- und anschließendem PUTX-Aufruf aktualisiert werden. (Dies ist nur im Locate Mode möglich.)

OVERLAP
Für ISAM-Dateien:
In Zusammenhang mit der Definition eines 2. Ein-/Ausgabebereichs im Programm (IOAREA2 im FCB) können Leseoperationen (GET/GETR) überlappend durchgeführt werden.
Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.
Bei NK-ISAM bedeutet „überlappende Verarbeitung“, dass benachbarte Blöcke ebenfalls in den ISAM-Pool eingelesen werden. OVERLAP=YES sollte nur bei vorwiegend sequenziellem Lesen genutzt werden.

= YES
Leseoperationen werden überlappend ausgeführt.

= NO
Leseoperationen werden nicht überlappend ausgeführt.

PAD = <integer 0..99>
Für ISAM-Dateien, die sequenziell erstellt werden (ISAM-Makroaufruf PUT):
Der „Blockfüllungsfaktor“ PAD gibt an, wie viel Platz im Datenblock für spätere Erweiterungen frei gehalten werden soll (in Prozent der mit BLKSIZE definierten Blocklänge). Die PAD-Angabe wirkt sich somit auf die Blocksplittingrate bei nichtsequenzieller Dateierweiterung aus. Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.
Die PAD-Angabe wirkt sich unterschiedlich aus bei NK-ISAM und K-ISAM. Bei NK-ISAM wird der Block mindestens bis zur PAD-Grenze gefüllt, bei K-ISAM höchstens bis zur PAD-Grenze.

POOLLNK = <name 1..8>
Nur ab VERSION=1 für ISAM-Dateien, die in Benutzer-ISAM-Pools verarbeitet werden (NK-ISAM):
„name“ ist der „Poolkettungsname“, der in die TFT eingetragen wird. Dieser Poolkettungsname muss mit ADDPLNK einem ISAM-Pool zugewiesen werden, der mit dem Makroaufruf CREPOOL erzeugt wurde. Zum OPEN-Zeitpunkt wird dieser Name an NK-ISAM übergeben; die Ein-/Ausgabepuffer der Datei werden in den zugehörigen ISAM-Pool übertragen.
Zulässiger Zeichenvorrat für „name“: Buchstaben, Ziffern und Sonderzeichen (entsprechend den Regeln zur Bildung von Namen).
Soll der Poolkettungsname über die Kommandoschnittstelle ansprechbar sein, muss er dem Datentyp <structured_name 1..8> entsprechen (siehe Handbuch „Kommandos“ [3]).

POOLSIZ = <integer 128..1048576>
Größe des dateispezifischen ISAM-Pools in Einheiten zu 2048 Byte.
Die Angabe bezieht sich nicht auf den mit POOLLNK angesprochenen ISAM-Pool.

PREFIX
wird nur in Verbindung mit MF=C oder MF=D ausgewertet und legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

= I
ist das voreingestellte Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen.

= *
Es wird kein Präfix generiert.

= <name 1..1>
Präfix, mit dem die generierten Feldnamen und Equates beginnen sollen.

RECFORM
gibt zum einen das Satzformat der mit „pfadname“ bezeichneten Datei an, zum anderen, welche Steuerzeichen bei der Ausgabe über einen Drucker zu berücksichtigen sind.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.
Das Satzformat wird bei den Zugriffsmethoden SAM und ISAM berücksichtigt. UPAM verarbeitet Dateien nur blockweise, eine RECFORM-Angabe wird ignoriert. BTAM ist zwar auch eine blockorientierte Zugriffsmethode, akzeptiert jedoch eine RECFORM-Angabe.
Die Satzformate sind im Kapitel „Zugriffsmethoden“ im Handbuch „Einführung in das DVS“ [1] detailliert beschrieben. Für den Zusammenhang zwischen RECFORM- und RECSIZE-Angabe siehe Operand RECSIZE; für die Drucksteuerzeichen-Auswertung wird auf das Kommando PRINT-DOCUMENT (Operand LINE-SPACING) in den Handbüchern „Kommandos“ [3] und „SPOOL“ [4] verwiesen.

= *BY-PROG
Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der RECFORM-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

= V
„pfadname“ besteht aus Sätzen variabler Länge, d.h. der Anwender muss bei der Programmierung berücksichtigen, dass den Datensätzen ein 4 Byte langes Feld vorangestellt wird, das in den Bytes 1-2 die Satzlänge als Binärzahl enthält. Die Bytes 3-4 werden vom System genutzt. Bei Eingabedateien wird das Satzlängenfeld vom System versorgt, bei Ausgabedateien muss der Anwender das Satzlängenfeld versorgen. Eine Angabe im RECSIZE-Operanden bezeichnet die maximale Satzlänge.
Für BTAM-Dateien wird die Angabe RECFORM=V wie RECFORM=U behandelt.

= F
„pfadname“ besteht aus Sätzen fester Länge, d.h. der Anwender braucht kein Satzlängen- und Steuerfeld zu berücksichtigen. Alle Sätze der Datei haben die gleiche Länge, die mit dem Operanden RECSIZE festgelegt wird (vgl. Seite 494f).

= U
„pfadname“ besteht aus Sätzen „undefinierter“ Länge; jeder Datenblock enthält nur einen Satz, dessen Länge bei der Eingabe vom System, bei der Ausgabe vom Anwender in einem Register übergeben wird (siehe Operand BLKSIZE, Seite 470).
RECFORM=U wandelt die Angabe LABEL=(STD,3) um in (STD,2).
Die Angabe RECFORM=U ist für ISAM-Dateien nicht zulässig.

= (...,N)
„pfadname“ ist keine Druckdatei, enthält also keine Drucksteuerzeichen und sollte nicht mit Steuerzeichenauswertung ausgedruckt werden.

= (...,M)
Das erste Datenbyte eines jeden Datensatzes wird als Steuerzeichen im EBCDI-Code interpretiert. Die Datei kann mit dem Kommando PRINT-DOCUMENT, Operand LINE-SPACING=*BY-EBCDIC-CONTROL ausgedruckt werden. Bei ISAM-Dateien wird der ISAM-Index berücksichtigt.

= (...,A)
Das erste Datenbyte eines jeden Datensatzes wird als ASA-Steuerzeichen interpretiert. Die Datei kann mit dem Kommando PRINT-DOCUMENT, Operand LINE-SPACING=*BY-ASA-CONTROL ausgedruckt werden.

RECSIZE
gibt die Satzlänge an, abhängig vom Wert des RECFORM-Operanden.
Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

= *BY-PROG
Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der RECSIZE-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

= <integer 0..32768>
für RECFORM=V: Die Angabe für RECSIZE wird ignoriert – abgesehen von folgender Ausnahme: Wird eine ISAM-Datei im Move Mode gelesen und ist der angegebene Wert für RECSIZE kleiner als der gelesene Satz, so wird der Satz nur in der Länge der RECSIZE-Angabe übertragen und die Fehlerbehandlung (DMS0AAD) eingeleitet.

für RECFORM=F: Satzlänge in Byte; alle Sätze der Datei sind gleich lang.

= <reg 2..12>
für RECFORM=U: mit dem Operanden RECSIZE muss ein Mehrzweckregister angegeben werden, das bei Ein- bzw. Ausgabe die aktuelle Satzlänge enthält. Bei der Eingabe wird das Register vom System versorgt, bei der Ausgabe muss es der Anwender versorgen.

RETPD = <integer 0..32767>
Der Benutzer kann mit „RETPD“ eine Schutzfrist (in Tagen) vereinbaren, während der kein Schreibzugriff (Ändern, Löschen) möglich ist.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

Ist die Schutzfrist abgelaufen, wird die Datei nicht automatisch gelöscht, es wird lediglich wieder Schreibzugriff zugelassen.

Die Schutzfrist kann auch über den CATAL-Makroaufruf beeinflusst werden: die RETPD-Angabe wird dann sofort in den Katalogeintrag übernommen; für Banddateien kann CATAL nur vor der ersten Dateieröffnung genutzt werden.

Für temporäre Dateien wird RETPD ignoriert.

SECLEV
Für Bandverarbeitung:
Der Operand SECLEV (Security Level) bezieht sich auf den TPIGNORE-Wert im Benutzerkatalog-Eintrag. Im Dialogbetrieb wird eine SECLEV-Angabe ignoriert. Im Stapelbetrieb können dazu berechtigte Benutzer mit SECLEV festlegen, ob Fehlermeldungen unterdrückt und/oder zusätzliche Kennsatzprüfungen durchgeführt werden sollen.

= HIGH
Fehlermeldungen werden im Stapelbetrieb an die Konsole ausgegeben. Läuft der Auftrag unter einer Benutzerkennung mit der Berechtigung TPIGNORE=YES im Benutzerkatalog-Eintrag, kann der Operator die Fehlermeldung ignorieren.

= LOW
nur für den Band-/Dateieigentümer zulässig, wenn im Benutzerkatalog-Eintrag TPIGNORE=YES definiert ist: im Stapelbetrieb werden bestimmte Fehlermeldungen unterdrückt.

= (...,OPR)
Die Angabe OPR (= Overwrite Protection) veranlasst das System zusätzliche Kennsatzprüfungen durchzuführen:
– wird eine Datei innerhalb eines Bandes im Anschluss an eine bereits existierende Datei erstellt, werden die Kennsätze der vorhergehenden Datei überprüft;
– das Freigabedatum der neuen Datei darf nicht höher sein als das der vorhergehenden.

SHARUPD
Für ISAM- oder UPAM-Plattendateien:
gibt an, ob mehrere Aufträge gleichzeitig die Datei in einem anderen Modus als OPEN INPUT eröffnen dürfen.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

= NO
Sobald die Datei von einem Auftrag mit OPEN ≠ INPUT eröffnet wird, wird sie für andere Aufträge gesperrt. Gleichzeitiger Zugriff mehrerer Aufträge auf die Datei ist nur möglich, wenn sie in allen Aufträgen als Eingabedatei verwendet wird, d.h. OPEN INPUT eröffnet ist. Ist die Datei bereits OPEN INPUT eröffnet, wird auch jeder Versuch, sie anders zu eröffnen, abgewiesen.

= YES
Für ISAM- oder PAM-Dateien:
Die Datei kann gleichzeitig von mehreren Aufträgen bearbeitet werden; es muss jedoch in allen Aufträgen SHARUPD=YES gelten. Bei UPAM kann der Anwender Datenblöcke, solange er sie verarbeitet, vor Zugriff durch andere Aufträge schützen. Bei ISAM werden Sperren – wenn nötig – vom System automatisch gesetzt, bei NK-ISAM als Satzsperren bzw. „Schlüsselsperren“, bei K-ISAM als Blocksperren. Bei NK-ISAM müssen Dateien, die für Shared-Update-Verarbeitung eröffnet werden, in hostspezifischen ISAM-Pools verarbeitet werden. Mit SHARUPD=YES ist für ISAM-Dateien gleichzeitig die WROUT-Funktion eingeschaltet (siehe Operand WROUT, Seite 512).

= WEAK
Nur ab VERSION=2, für UPAM-Verarbeitung:
garantiert Schreibsicherheit, aber nicht Lesesicherheit. Nur ein Auftrag kann die Datei zur Aktualisierung eröffnen, andere Aufträge können sie aber gleichzeitig als Eingabedatei nutzen. Der Benutzer muss in seinem Programm berücksichtigen, dass sich der Inhalt der Datei ändern kann, während er sie als Eingabedatei verwendet.

SPACE

Nur für Plattendateien: beeinflusst über Primär-, Sekundär- oder Absolutzuweisung die Speicherplatzbelegung der Datei „pfadname“. Der SPACE-Operand wird immer ausgewertet, auch ohne LINK-Operanden im aktuellen Makroaufruf. Primär, Sekundär- und Absolutzuweisung sind im Abschnitt „Anfordern von Speicherplatz“, Handbuch „Einführung in das DVS“ [1] ausführlich beschrieben.

Voreinstellung: Ist SPACE nicht angegeben, so gilt:

- Hat die Datei bereits Speicherplatz, so bleiben ihr Speicherplatz und ihre Sekundärzuweisung unverändert.
- Andernfalls werden für Primär- und Sekundärzuweisung Voreinstellungen wirksam. Für Dateien auf Net-Storage werden diese Voreinstellungen von der Speicherplatzverwaltung fest vorgegeben. Für andere Plattendateien ergeben sich diese Voreinstellungen aus Werten, die vom Systemverwalter einstellbar sind.

Für eine geöffnete oder eine von einer fremden Task exklusiv reservierte Datei wird ein FILE-Makroaufruf mit SPACE-Operand zurückgewiesen. Auch Schutzattribute einer Datei oder Dateigenerationsgruppe werden berücksichtigt. Fordert der Anwender mehr Platz im Pubset an, als ihm entsprechend Eintrag im Benutzerkatalog zugestanden ist, wird der FILE-Makroaufruf zurückgewiesen. Ist der Anwender berechtigt, sein Speicherplatz-Kontingent zu überschreiten, informiert ihn das System mit einer Meldung darüber.

Um den Verwaltungsaufwand des Systems und den Speicherplatzverbrauch gering zu halten, ist bei der Definition vom Primär- und Sekundärzuweisung Folgendes zu beachten:

- die Primärzuweisung sollte der erwarteten Größe der einzurichtenden Datei entsprechen
- die Sekundärzuweisung sollte dem erwarteten Wachstum der einzurichtenden Datei entsprechen.

Bei der Eröffnung einer Datei mit BLKSIZE=(STD,n), wobei n ≥ 2, muss für die Anzahl „p“ der für sie reservierten PAM-Seiten und für ihre Sekundärallokierung „s“ gelten:

	SPACE-Operand	
Dateityp	**p**	**s**
SAM	≥ 2n	≥ n
ISAM: K-ISAM NK-ISAM	> n > n	
PAM (gekettete Ein-/Ausgabe)	> 0	

= <integer -2147483647..2147483647>
Primärzuweisung, die sofort in Kraft tritt

Im Folgenden bezeichnet k die Anzahl der PAM-Blöcke pro Unit (kleinste Verwaltungseinheit der Speicherplatzverwaltung für Plattendateien; zu Unit siehe Abschnitt „Anfordern von Speicherplatz“, Handbuch „Einführung in das DVS“ [1]).

1..2147483647:
Die Speicherplatzzuweisung wird auf ein Vielfaches von k aufgerundet und die entsprechende Zahl PAM-Seiten auf dem Pubset oder der Privatplatte zugewiesen.
Der Anwender sollte beachten, dass jeder FILE-Aufruf mit positiver Primärzuweisung für die Datei Speicherplatz reserviert. Auf diese Weise ist bei hoher Primärzuweisung das Speicherplatzkontingent des Anwenders schnell erschöpft.

Für Dateien auf gemeinschaftlichen Platten und auf Privatplatten werden die Platten für die Speicherplatzzuweisung wie folgt bestimmt:
- Ist der Operand VOLUME nicht angegeben und belegt die Datei noch keinen Speicherplatz, so wird der Speicherplatz auf gemeinschaftlichen Platten zugewiesen. Für Dateien auf SM-Pubsets werden hierbei bei der Selektion des Volume-Sets berücksichtigt:
  - vorläufiges Dateiformat
  - Angaben zu AVAIL, WORKFIL, VOLSET, IOPERF, IOUSAGE, DISKWR
  - Eigenschaft permanent/temporär
  - Storage-Klasse, die der Datei zugewiesen wird.
- Ist der Operand VOLUME nicht angegeben und belebt die Datei bereits Speicherplatz, so werden nach Möglichkeit die bisher von der Datei schon belegten Platten zur Speicherplatzzuweisung herangezogen.
- Ist der Operand VOLUME angegeben, beginnt die Speicherplatzzuweisung auf der ersten über den Operanden VOLUME enthaltenen Platte. Falls diese nicht ausreicht, wird mit der zweiten über den Operanden VOLUME erhaltenen Platte weitergemacht usw.

Eine Speicherplatzzuweisung auf einem Pubset wird abgewiesen, falls auf dem Pubset insgesamt weniger Seiten frei sind, als in der Primärzuweisung angegeben wurden.

Für Dateien auf gemeinschaftlichen Platten und auf Privatplatten wird eine Teilzuweisung vorgenommen, falls der Operand VOLUME angegeben ist und die über den Operanden VOLUME erhaltenen Platten zusammen weniger freie PAM-Seiten enthalten, als in der Primärzuweisung angegeben wurden (jedoch mindestens eine freie Unit). Bei Angabe von gemeinschaftlichen Platten wird allerdings der FILE-Aufruf abgewiesen, wenn auf dem gesamten Pubset weniger PAM-Seiten frei sind, als in der Primärzuweisung angegeben wurden (siehe oben).

Der Eintrag einer Benutzerkennung im Benutzerkatalog enthält deren Kontingent an gemeinschaftlichem Speicherplatz. Wird dieses durch die Speicherplatzanforderung für die Benutzerkennung der Datei überschritten, so gilt: Hat die Benutzerkennung der

Datei laut Eintrag im Benutzerkatalog die Berechtigung zur Überschreitung des Kontigents, so wird die aufrufende Task mit einer Warnung von der Überschreitung des Kontingents informiert, andernfalls wird der FILE-Aufruf abgewiesen.

Bei Privatplatten wird eine Teilzuweisung vorgenommen (falls möglich), wenn die Anforderung das freie Speicherplatzkontingent überschreitet.

Würde eine Speicherplatzanforderung zum Überschreiten der maximalen im Katalogeintrag darstellbaren Dateigröße (=16777215 PAM-Seiten) führen, so wird nur die maximal mögliche Teilzuweisung vorgenommen.

-2147483647..-1:
Speicherplatz-Freigabe nach Rundung der Primärzuweisung auf ein Vielfaches von k. Die Speicherplatzfreigabe erfolgt entsprechend der Datenträgerliste vom Dateiende in Richtung Dateianfang (Angaben im VOLUME-Operanden werden ignoriert). Es werden nur „unbeschriebene“ Units freigegeben, für ISAM-Dateien können Index- und Datenteil nicht getrennt freigegeben werden (siehe Operand DSPACE, Seite 480).

Falls die Datei nach der Speicherplatzfreigabe keinen Speicherplatz mehr belegt, werden im Katalogeintrag gelöscht: AVAIL, WORKFIL (nicht jedoch für Generationen), STOCLAS, Indikator „Datei enthält defekten Block“, Indikator „S0-Migration verboten“, vorläufiges Dateiformat.

Wenn eine BS2000-Datei auf einem Net-Storage-Volume nach der Speicherplatzfreigabe keinen Speicherplatz mehr hat, dann existiert sie auf dem Net-Storage-Volume nicht mehr. In ihrem Katalogeintrag werden alle Hinweise auf das Net-Storage-Volume entfernt.

Bei Dateien auf Privatplatten bleiben mindestens 3 PAM-Seiten und bei Node-Files bleiben mindestens 4 PAM-Seiten zugewiesen. Bei existierenden Dateien mit BLKSIZE=(STD,k) bleiben mindestens so viele PAM-Seiten zugewiesen, wie zur Dateieröffnung nötig sind. Die Anzahl der verbleibenden PAM-Seiten wird hierbei von der Speicherplatzverwaltung festgelegt.

Gilt DESTROY=YES im Katalogeintrag, werden alle freigegebenen PAM-Seiten mit X'00' überschrieben (ohne Beachtung einer Unit-Grenze). Hierbei werden ggf. benötigte Privatplatten angefordert. Bei einer mit DESTROY=NO katalogisierten Datei erfolgt dieses Überschreiben nur dann, wenn der Destroy-Level (Klasse-2-Systemparameter DESTLEV) genügend hoch eingestellt ist.

0:
keine Veränderung bzgl. der Speicherplatzreservierung; für Dateien auf Privatplatte nur zulässig, wenn die Datei bereits Speicherplatz belegt. Eine gleichzeitige Angabe von VOLUME wird ignoriert, falls die Datei bereits Speicherplatz belegt, andernfalls abgewiesen.

= (<integer -2147483647..2147483647>,<integer 1..32767>)
legt Primär- und Sekundärzuweisung fest. Im Gegensatz zur Primärzuweisung wird die Sekundärzuweisung nicht sofort bei Eingabe des FILE-Makroaufrufs wirksam, sondern erst, wenn bei Dateierstellung oder -erweiterung der reservierte Speicherplatz nicht ausreicht. Der Wert der Sekundärzuweisung wird in den Katalogeintrag übernommen (Feld S-ALLOC der Ausgabe des Kommandos SHOW-FILE-ATTRIBUTES).

<integer -2147483647..2147483647>:
siehe oben: Primärzuweisung

<integer 1..32767>:
Sekundärzuweisung, d.h. die Anzahl PAM-Seiten, um die der Speicherplatz bei Bedarf erweitert werden soll. Die Sekundärzuweisung wird unverändert in den Katalogeintrag übernommen. Erst wenn die Sekundärzuweisung in Kraft tritt, wird sie auf ein Vielfaches von k aufgerundet.

Mit SPACE=(0,<integer 1..32767>) wird die Sekundärzuweisung festgelegt/geändert und der (neue) Wert in den Katalogeintrag übernommen. Diese Angabe ist für Dateien/Generationen auf Privatplatte nur zulässig, wenn vorher bereits Speicherplatz für diese Datei oder Dateigeneration angefordert wurde.

(...,0):
verhindert dynamische Erweiterung der Datei.

= (<integer -2147483647..2147483647>[,<integer 1..32767>],*KEEP)
Nur ab VERSION=2 bei Speicherplatz-Freigabe für eine Datei auf gemeinschaftlichen Datenträgern oder Net-Storage-Volumes:
„*KEEP“ bedeutet, dass der Datei mindestens eine Allokierungs-Einheit zugewiesen bleibt.

= (<integer -2147483647..2147483647>,<integer -2147483647..2147483647>,ABS)
Absolutzuweisung (nur zusammen mit VOLUME). Reicht der freie Speicherplatz auf der Platte nicht aus, wird der FILE-Makroaufruf abgewiesen; es erfolgt keine Teilzuweisung. Da die Absolutzuweisung sich immer nur auf einen Datenträger bezieht, muss für jeden Datenträger ein separater FILE-Makroaufruf gegeben werden.

Ist die Absolutzuweisung die erste Speicherplatzanforderung für eine Datei, erhält die Sekundärzuweisung den Wert 0. Angegeben werden:

1. Blocknummer der PAM-Seite, auf der die Speicherplatzreservierung auf der Privatplatte beginnen soll. Da Speicherplatz nur in Units reserviert wird, gilt für „seite“: seite = k*n + 1 ($n \geq 0$). Auf welcher PAM-Seite die Speicherplatzreservierung einer Platte beginnen kann, hängt von der Initialisierung der Platte ab.

2. Angabe, wie viele PAM-Seiten auf dem Datenträger reserviert werden sollen; muss ein Vielfaches von k sein. Da die Kapazität von Plattenspeichern von Plattentyp und Initialisierung der Platte abhängt, müssen die Maximalwerte beim Systemverwalter erfragt werden. Die Obergrenze für diese Maximalwerte beträgt 2147483647 (wie bei der Primärzuweisung).
3. *ABS:* Das Schlüsselwort „ABS“ kennzeichnet die Absolutzuweisung.

Absolutzuweisung für eine Datei auf einem Net-Storage-Volume ist nicht möglich.

STATE = FOREIGN

Für Dateien auf privaten Datenträgern oder auf Net-Storage-Volumes, die keinen Eintrag im Systemkatalog besitzen, wird ein Katalogeintrag erstellt (Datei-Import). Für Dateigenerationen muss gegebenenfalls vorher der Gruppeneintrag rekonstruiert werden (mit CATAL-Makroaufruf). Dateien, die mit STATE=FOREIGN übernommen werden, sollten aus dem Katalog ihres „alten“ Eigentümers exportiert werden (ERASE CATALOG).

Im Operanden VOLUME müssen die Archivnummern der für die Verarbeitung der Datei benötigten Datenträger in der richtigen Reihenfolge aufgelistet sein. Falls MAREN zur Verfügung steht und die Datenträger der Datei im MAREN-Katalog stehen, kann die VOLUME-Angabe auch weggelassen werden; MAREN liefert dann die Archivnummern.

Folgende Angaben dürfen nicht zusammen mit STATE=FOREIGN gemacht werden: DEVICE=WORK, TVSN, TSET, VSEQ.

Datei auf Privatplatte:
Der Katalogeintrag wird aus dem F1-Label der ersten im Operanden VOLUME angegebenen bzw. über MAREN erhaltenen Privatplatte erstellt. Die Datei kann nur auf die im F1-Kennsatz enthaltene Benutzerkennung importiert werden. Die Datei kann auch auf andere Pubsets importiert werden, als auf den, auf dem sie erstmals katalogisiert war. Eine im F1-Kennsatz als mehrbenutzbar katalogisierte Datei kann von jeder Task (d.h. unabhängig von der Benutzerkennung der Task) auf die im F1-Kennsatz enthaltene Benutzerkennung importiert werden.

Dateien auf Net-Storage:
Der Katalogeintrag wird erstellt aus dem Katalog auf dem Net-Storage-Volume, das im Operanden VOLUME angegeben ist. Die Datei kann nur auf den Pubset importiert werden, der dem im Operanden VOLUME angegebenen Net-Storage-Volume zugeordnet ist.

Banddateien:
Dateimerkmale einer FOREIGN-Datei können nicht mit CATAL verändert werden.

Hat die FOREIGN-Banddatei Standardkennsätze, werden zum OPEN-Zeitpunkt die Dateimerkmale RECFORM, RECSIZE, BLKSIZE und CODE aus dem HDR2-Kennsatz in den Katalog übernommen. Die Datei kann unter mehreren Benutzerkennungen katalogisiert sein; das System sorgt dann für Konsistenz zwischen dem Katalogeintrag und der Kennsatzinformation.

Hat die FOREIGN-Datei NSTD-Kennsätze oder keine Standardkennsätze, muss der Anwender im FILE-Makroaufruf die Operanden RECFORM, RECSIZE und BLKSIZE versorgen. Wird die Datei unter mehreren Benutzerkennungen katalogisiert, ist jeder Anwender selbst für die Konsistenz zwischen Katalogeinträgen und Kennsatzinformation verantwortlich.

Für die Übernahme einer FOREIGN-Banddatei mit Standardkennsätzen gilt: Ist der Anwender nicht Dateieigentümer, müssen Datenträger und Datei mehrbenutzbar sein (Kennzeichen in VOL1-, HDR1-Kennsatz);

STOCLAS

Nur für VERSION=3:

Beim Anlegen einer Datei auf einem SM-Pubset kann der Datei eine Storage-Klasse zugewiesen werden. Diese enthält dann die Attribute, denen der Ablageort der Datei genügen soll. Falls der Storage-Klasse eine Volume-Set-Liste zugeordnet ist, wird die Datei vorrangig auf einem Volume-Set dieser Liste angelegt.

Eine Storage-Klassen-relevante Angabe liegt in folgenden Fällen vor:

– wenn einer der Operanden AVAIL, DISKWR, VOLUME, VOLSET oder WORKFIL angegeben ist.
– wenn für den Operanden DEVICE ein Wert ungleich NETSTOR und ungleich STDDISK angegeben ist.
– wenn für einen der Operanden IOPERF oder IOUSAGE ein Wert ungleich Null-Operand angegeben ist.

Im Eintrag jeder Benutzerkennung im Benutzerkatalog eines SM-Pubsets kann eine Default-Storage-Klasse hinterlegt sein. Sie kann mit SHOW-USER-ATTRIBUTES INF=PUBSET-ATTR angezeigt werden.

Beim Anlegen einer Datei oder Dateigeneration auf einem SM-Pubset auf einer Benutzerkennung mit Default-Storage-Klasse an dem SM-Pubset gilt: Wenn kein Recht auf physikalische Allokierung vorliegt und die Datei nicht auf einem Volume-Set für Arbeitsdateien angelegt wird, werden Storage-Klassen-relevante Angaben und STOCLAS=*NONE unwirksam, d.h. sie werden ignoriert, sofern sie nicht abgewiesen werden. (IOPERF und IOUSAGE werden trotzdem in den TFT-Eintrag eingetragen.)

Beim Anlegen einer Datei (nicht Dateigeneration) auf einem SM-Pubset auf einer Benutzerkennung mit Default-Storage-Klasse an dem SM-Pubset wird der Datei die benutzerspezifische Default-Storage-Klasse zugewiesen, wenn weder eine Angabe zu STOCLAS noch eine Storage-Klassen-relevante Angabe wirksam ist.

In einem FGG-Index kann ebenfalls eine Default-Storage-Klasse hinterlegt sein. Diese wird beim Anlegen einer Dateigeneration auf einem SM-Pubset zugewiesen, wenn weder eine Angabe zu STOCLAS noch eine Storage-Klassen-relevante Angabe wirksam ist.

Wenn im Eintrag der Benutzerkennung der Datei bzw. im FGG-Index eine Default-Storage-Klasse hinterlegt ist, die am betroffenen SM-Pubset nicht existiert oder auf die der Aufrufer kein Zugriffsrecht hat, so muss zum Anlegen der Datei bzw. Dateigeneration im Operanden STOCLAS der Wert *NONE oder eine andere Storage-Klasse angegeben werden.

Wenn die Datei auf einem SF-Pubset oder auf einem privaten Datenträger angelegt wird, wird der Datei auch bei Angabe eines Storage-Klassen-Namens und auch beim Vorhandensein einer Default-Storage-Klasse keine Storage-Klasse zugewiesen.

Eine Storage-Klasse kann auch zugewiesen werden, wenn eine Datei auf einem Net-Storage-Volume angelegt wird; hierbei kann keine Arbeitsdatei entstehen und auch keine Datei mit PAM-Key.

= *NONE
Der Datei wird keine Storage-Klasse zugewiesen, eine evtl. vorhandene Default-Storage-Klasse wird nicht herangezogen.

= <c-string 1..8>
Name der Storage-Klasse, die der Datei zugewiesen wird. Die Angabe wird in folgenden Fällen abgewiesen:
– die Datei belegt bereits Speicherplatz
– SPACE ist mit nichtpositiver Primärzuweisung angegeben
– eine Storage-Klassen-relevante Angabe liegt vor
– die Storage-Klasse existiert nicht am betroffenen SM-Pubset
– der Aufrufer hat kein Zugriffsrecht auf die Storage-Klasse.

STREAM
Für BTAM-Banddateien:
Ermöglicht Ein-/Ausgaben im Streaming-Modus. Das bedeutet einerseits, dass die im MAV-Modus (Operanden BTAMRQS im FCB- und REQNO im BTAM-Aufruf) angebotenen geketteten Ein-/Ausgabe-Aufträge (Operand CHAINIO) ihrerseits verkettet werden. Andererseits bedeutet es, dass im Falle eines Streaming-Bandgerätes der Modus 'streamen' hardwaremäßig eingestellt werden soll.

= <u>NO</u>
Streaming-Modus nicht einschalten, sofern nicht im FCB des Programms STREAM= YES festgelegt ist.

= YES
Streaming-Modus einschalten.

TAPEWR
Nur ab VERSION=1, für Dateien auf Magnetbandkassetten:
der Anwender kann bestimmen, ob die Ausgabe gepuffert erfolgen soll.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

= DEVICE-BUFFER
Die Ausgabe wird über die Gerätesteuerung gepuffert, wodurch eine hohe Datenübertragungsrate erzielt werden kann.

= IMMEDIATE
Die Ausgabe wird nicht gepuffert.

TPMARK
Für Banddateien ohne Standardkennsätze (LABEL=NO/NSTD):
legt fest, ob beim Erzeugen einer Banddatei Abschnittsmarken geschrieben werden. Dateien mit LABEL=(STD,n) erhalten standardmäßig Abschnittsmarken nach den Kennsätzen.

= NO
Es wird für Banddateien ohne Standardkennsätze keine Abschnittsmarke geschrieben, sofern nicht im FCB des Programms TPMARK=YES festgelegt ist.

= YES
Banddateien mit NSTD-Kennsätzen: die Abschnittsmarke folgt dem Kennsatz.

Banddateien ohne Kennsätze: die Abschnittsmarke wird an den Beginn des Bandes geschrieben.

TRANS
Für Banddateien, die als Eingabedateien genutzt werden und die nicht mit CODE=EBCDIC erstellt wurden:
legt fest, wie der Code der Datei beim Lesen umgesetzt werden soll.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

= YES
ISO-7-Bit-Code oder OWN-Code werden in EBCDI-Code umgesetzt.

= NO
ISO-7-Bit-Code wird mit einer führenden Null in ein 8-Bit-Format umgesetzt.

TSET

Für Ausgabe-Banddateien mit Standardkennsätzen:
legt über einen TST-Eintrag (in der Tape Set Table) eine Bandmenge für die Datei fest oder stellt die Verbindung zu einem bestehenden TST-Eintrag her, sofern der LINK-Operand angegeben ist. Der entsprechende TFT-Eintrag verweist dann auf den damit verknüpften TST-Eintrag.

Ein TST-Eintrag besteht im Wesentlichen aus einem TSET-Namen und aus einer Datenträgerliste, die mit dem VOLUME-Operanden definiert oder erweitert werden kann. Mit nachfolgenden FILE-Aufrufen kann man sich durch Angabe des TSET-Namens auf diese Datenträgerliste beziehen und sie ggf. erweitern.

Falls TSET zusammen mit LINK angegeben ist, gilt:
– Ist der aktuelle TSET-Name bisher in keinem TST-Eintrag enthalten, wird ein neuer Eintrag mit TSET-SHR=1 angelegt (TSET-SHR bezeichnet die Anzahl der mit diesem TST-Eintrag verbundenen TFT-Einträge; siehe Ausgabe des Kommandos SHOW-FILE-LINK).
– Existiert bereits ein TST-Eintrag gleichen Namens und ist ein in der TFT nicht vorhandener Dateikettungsname angegeben, wird TSET-SHR um 1 erhöht.
– Beim Freigeben eines TFT-Eintrags, der mit einem TST-Eintrag verknüpft ist, wird dessen TSET-SHR um 1 vermindert. Außerdem wird der TST-Eintrag freigegeben, wenn dabei TSET-SHR=0 erreicht wird.

Folgende Bedingungen müssen bei Angabe von TSET erfüllt sein:
– Falls eine bereits katalogisierte Datei angegeben ist, muss die Volume-Table im Katalogeintrag leer sein.
– Falls eine neue Datei angegeben ist, muss der Operand DEVICE angegeben sein.
– Falls der Operand DEVICE angegeben ist, muss sein Wert ein Bandtyp sein.

Folgende Operanden dürfen nicht zusammen mit TSET angegeben werden:
STATE=FOREIGN, DEVICE=WORK, VSEQ, TVSN, FSEQ=UNK, FSEQ=n mit n>1, FSEQ=Null-Operand, VOLUME=REMOVE-UNUSED

= <name 1..4>
Tape-Set-Name in dem TST-Eintrag, auf den Bezug genommen wird. Falls der TST-Eintrag noch nicht existiert oder noch keinen File-Set-Identifier hat und der VOLUME-Operand angegeben ist, wird die erste über den VOLUME-Operanden erhaltene VSN als File-Set-Identifier eingetragen.

= (<name 1..4>,<name 1..6>)
der vierstellige Name bezeichnet einen TST-Eintrag, der sechsstellige Name (Archivnummer „vsn“) ist das Dateimengenkennzeichen (File Set Identifier).

Falls der TST-Eintrag bereits existiert, muss der File-Set-Identifier in der TSET-Angabe mit dem File-Set-Identifier im TST-Eintrag übereinstimmen. Bei Dateieröffnung müssen die Dateimengenkennzeichen im TST-Eintrag und im HDR1-Kennsatz übereinstimmen.

TVSN
Nur für Banddateien, die als Eingabedateien genutzt werden:
gibt eine temporäre Liste von Archivnummern für die Verarbeitung an. Diese bildet dann die Datenträgerliste. Wird der Operand TVSN angegeben, wird bei der Dateiverarbeitung die Datenträgerliste des Katalogeintrags ignoriert; es werden nur die mit TVSN angegebenen Datenträger verwendet. Der Katalogeintrag wird jedoch nicht verändert.

Folgende Operanden dürfen nicht zusammen mit TVSN angegeben werden:
*DUMMY, STATE=FOREIGN, TSET, VOLUME

= <@adr>
Nur ab VERSION=2:
adr ist eine symbolische Adresse im Programm, an der mit dem Makroaufruf FILELST TVSN=... eine TVSN-Liste angelegt wurde.
Das Zeichen „@“ ist Bestandteil des Operandenwertes und muss mit angegeben werden.

= list-poss (255): <name 1..6>
gibt die Archivnummern der Datenträger an, die für die Eingabe benötigt werden.

VALLEN = <integer 0..255>
Für K-ISAM-Dateien:
legt die Länge einer Wertmarkierung (Value Flag) im ISAM-Index fest.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

Wertmarkierungen werden bei K-ISAM blockweise ausgewertet und entsprechend der VALPROP-Angabe in den nächsthöheren Indexeintrag übernommen; bei NK-ISAM wird die VALLEN-Angabe ignoriert.

= *BY-PROG
Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der VALLEN-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

VALPROP
Für K-ISAM-Dateien:
VALPROP (= Value Propagation) legt fest, wie die Wertmarkierung in die Indexeinträge zu übernehmen ist.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

= *BY-PROG
Nur ab Version=3 und nur relevant bei Angabe des Operanden DATATTR:
Der VALPROP-Wert aus dem Katalogeintrag der Referenzdatei wird ignoriert.

= MIN
Der niedrigste Wert für die Wertmarkierung innerhalb eines Daten- oder Indexblocks wird in den Indexeintrag der nächsthöheren Stufe übernommen.

= MAX
Der höchste Wert der Wertmarkierung innerhalb eines Daten- oder Indexblocks wird in den Indexeintrag der nächsthöheren Stufe übernommen.

VERSION
legt fest, welche Version der Operandenliste und welcher SVC erzeugt werden soll.

Voreinstellung: VERSION=0

Hinweis

In allen FILE-Aufrufen, die sich durch den MF-Operanden unterscheiden (MF=L/E/D/C), muss der Operand VERSION den gleichen Wert haben.

= 0
Es wird das „alte“ Format der Operandenliste (Stand BS2000 V9.0) (SVC 159) erzeugt. Zur Unterstützung von Gerätetypen und Funktionen, die erst nach dieser Version eingeführt wurden muss daher ein neues Format der Operandenliste (ab VERSION=1) verwendet werden.

= 1
ist gültig ab BS2000 V9.5: Es wird eine Operandenliste mit Standardheader (SVC 144) erzeugt. Über die Operanden der BS2000-Version 9.0 hinaus unterstützt dieses Format auch die zur BS2000-Version 9.5 neu eingeführten Operanden BLKCTRL, CLOSMSG, DESTOC, POOLLNK und TAPEWR sowie die ab V9.5 gültigen Gerätetypen.

= 2
ist gültig ab BS2000 V10.0:
Es wird eine Operandenliste mit Standardheader (SVC 144) erzeugt. Im Unterschied zur Operandenliste mit VERSION=1 sind die variablen Teile (für die Operanden VOLUME, DVOLUME, TVSN, MOUNT und VSEQ) der Operandenliste in eigene Bereiche ausgelagert. Diese Bereiche können mit dem Makro FILELST erzeugt werden.

Folgende Operanden bzw. Operandenwerte werden nur ab VERSION=2 unterstützt: IOPERF, IOUSAGE, CLOSE, BLKCTRL=DATA2K/DATA4K, SHARUPD=WEAK und SPACE=(...,*KEEP)

= 3
ist gültig ab BS2000/OSD-BC V3.0. Es wird eine Operandenliste mit Standardheader (SVC 144) erzeugt.

Folgende Operanden bzw. Operandenwerte werden nur ab VERSION=3 unterstützt: AVAIL, CLOSE=KEEP-DATA-IN-CACHE, CODE=ISO7D, DATATTR, DISKWR, EXC32GB, LOCKENV, NFTYPE, POOLSIZ, STOCLAS, VOLSET, VOLUME= REMOVE-UNUSED, WORKFIL und bei einigen Operanden der Wert =*BY-PROG.

Dieses Format der Operandenliste bietet die beste Unterstützung bei der Programmierung; es sollte auch im Hinblick auf die Weiterentwicklung des FILE-Makros verwendet werden.

VOLSET
Nur ab VERSION=3 für Dateien auf SM-Pubsets:
legt fest, auf welchem Volume-Set die Datei angelegt werden soll. Die Angabe wird abgewiesen, wenn die Datei bereits Speicherplatz belegt, die Datei auf einem Net-Storage-Volume angelegt werden soll oder wenn SPACE mit nichtpositiver Primärzuweisung angegeben ist.

Die Angabe eines Volume-Sets mit permanenter Datenhaltung erfordert das Recht auf physikalische Allokierung.

= <c-string: catid 1..4>
Katalogkennung des Volume-Sets, auf dem die Datei angelegt werden soll.

= *CONTROL
Die Datei wird auf dem Control-Volume-Set des SM-Pubsets angelegt.

VOLUME
gibt an, welche Datenträger für die Dateiverarbeitung benötigt werden.
Sind beim Neueinrichten einer Datei weder DEVICE noch VOLUME angegeben, wird die Datei auf gemeinschaftlichen Datenträgern angelegt.

Net-Storage-Volumes gelten als Platten und können ohne das Recht auf physikalische Allokierung angegeben werden.

Falls die erste über den Operanden VOLUME erhaltene VSN ein Net-Storage-Volume bezeichnet, das der Pubset zugeordnet ist, auf dem die Datei liegt bzw. angelegt werden soll, wird der VSN auch dann das Net-Storage-Volume zugeordnet, wenn die VSN gleichzeitig eine Privatplatte bezeichnet.

Für Dateien auf gemeinschaftlichen Platten, Net-Storage-Volumes oder auf Privatplatten:
Die Datenträgerliste enthält alle Platten, auf denen (ggf. nach Abschluss der Speicherplatzzuweisung) Extents der Datei liegen.
Das DVS versucht, den gesamten mit SPACE angeforderten Platz auf der ersten Platte zu reservieren. „Nicht verwendete“ Archivnummern werden für spätere Erweiterungen in die Datenträgerliste des Katalogeintrags übernommen.
Falls keine Speicherplatzzuweisung nötig ist, werden die angegebenen Archivnummern ignoriert.

Für Dateien auf einem Net-Storage-Volume:
Die Datenträgerliste besteht aus der VSN des Net-Storage-Volumes auf dem die Datei liegt bzw. angelegt wird. Die Volume-Table im Katalogeintrag und (bei Angabe des Operanden LINK) die Volume-Table im erzeugten TFT-Eintrag bestehen ebenfalls aus dieser einen VSN. Der Pubset, an dem die Datei katalogisiert ist bzw. katalogisiert wird, muss dem Net-Storage-Volume zugeordnet sein, auf dem die Datei liegt bzw. angelegt wird.

Für Banddateien:
Die Datenträgerliste setzt sich zusammen aus den Archivnummern des Katalogeintrags (sofern bereits vorhanden) und den sich logisch daran anschließenden Archivnummern des VOLUME-Operanden. Es wird standardmäßig der erste Datenträger aus der Datenträgerliste angefordert (sofern nicht MOUNT=0 angegeben wird). Fordert der Anwender die Bereitstellung mehrerer Datenträger, ist im MOUNT-Operanden anzugeben, wie viele Datenträger gleichzeitig bereitgestellt werden sollen.

Ist „Pfadname“ noch nicht katalogisiert, werden die Archivnummern des VOLUME-Operanden in den Katalogeintrag übernommen. Außerdem kann mit dem TSET-Operanden die Verbindung zu einem TST-Eintrag hergestellt werden.

Die Auswirkungen des Operanden VOLUME hängen davon ab, ob der Operand TSET angegeben wird. Ist er nicht angegeben, wird die Datenträgerliste unverändert in den Katalogeintrag übernommen. Ist TSET angegeben, wird zunächst die Datenträgerliste des TST-Eintrages aktualisiert bzw. erstellt und anschließend die Datenträgerliste des Katalogeintrages entsprechend dem TST-Eintrag erstellt. Nach Dateieröffnung wird dann der Katalogeintrag anhand der Datenträgerliste im TST-Eintrag aktualisiert.

Ist „Pfadname“ bereits katalogisiert, bilden die Archivnummern des VOLUME-Operanden eine Erweiterung der Volume-Table des Katalogeintrags. Der VOLUME-Operand darf also keine Archivnummern enthalten, die bereits im Katalogeintrag enthalten sind.

= @adr
Nur ab VERSION=2:
adr ist eine symbolische Adresse im Programm, an der mit dem Makroaufruf FILELST VOLUME=... eine VOLUME-Liste angelegt wurde.
Das Zeichen „@“ ist Bestandteil des Operandenwertes und muss mit angegeben werden.

= REMOVE-UNUSED
Nur für bereits katalogisierte Banddateien:
Es werden alle Bänder aus der Volume-Table des Katalogeintrags entfernt, auf denen keine Daten der Datei liegen.
LINK und TSET dürfen nicht zusammen mit REMOVE-UNUSED angegeben werden.

= PRIVATE
Für die Dateiverarbeitung wird ein privater Datenträger benötigt. Dazu wird der Operator durch eine Meldung an der Konsole aufgefordert, die Archivnummer des Datenträgers anzugeben.
In einem FILE-Aufruf für die Pseudodatei *DUMMY wird VOLUME=PRIVATE ignoriert.

= (PRIVATE,<integer 1..9>)
Für die Dateiverarbeitung werden mehrere private Datenträger benötigt. Dazu wird der Operator durch eine Meldung an der Konsole aufgefordert, die Archivnummern der Datenträger anzugeben.
Für die Pseudodatei *DUMMY wird VOLUME=(PRIVATE,<integer 1..9>) ignoriert.

= list-poss(255): <name 1..6>
Archivnummern (VSNs) der angeforderten Datenträger

VSEQ
Für katalogisierte Banddateien mit Standardkennsätzen:
Der VSEQ-Operand ermöglicht eine abschnittsweise Dateiverarbeitung. Ein Dateiabschnitt ist der Teil einer Mehrbanddatei, der sich auf einem Band befindet (für den Einfluss auf den Aufbau der TFT-Datenträgerliste siehe „Hinweise zur Programmierung“ auf Seite 513).

Der VSEQ-Operand bezieht sich auf die Datenträgerliste (siehe Operand VOLUME).
Die Dateiabschnittsnummern entsprechen relativen Archivnummern, d.h. sie geben die Position der Archivnummer in der Datenträgerliste an.

Einzelangabe: Wird im VSEQ-Operanden nur eine Dateiabschnittsnummer angegeben, so werden alle Datenträger ab dem so bezeichneten Eintrag in die Volume-Table des TFT-Eintrags übernommen.

Listenangabe: Wird im VSEQ-Operanden eine Liste von Dateiabschnittsnummern angegeben, so werden die so bezeichneten Einträge in die Volume-Table des TFT-Eintrags übernommen.

VSEQ darf nicht zusammen mit TSET oder STATE=FOREIGN angegeben werden. Wurde die Datei bisher nicht bzw. nur mit CATAL katalogisiert, wird jede VSEQ-Angabe außer VSEQ=1 abgewiesen.

= @adr
Nur ab VERSION=2:
adr ist eine symbolische Adresse im Programm, an der mit dem Makroaufruf FILELST VSEQ=... eine VSEQ-Liste angelegt wurde.
Das Zeichen „@“ ist Bestandteil des Operandenwertes und muss mit angegeben werden.

= <integer 1..255>
Nummer des Abschnitts, an dem die Verarbeitung beginnen soll.

Ist „pfadname“ eine Ausgabedatei (OPEN=OUTPUT/OUTIN), muss VSEQ=1 angegeben werden.

Ist „pfadname“ eine Eingabedatei, bezeichnet VSEQ=zahl den Dateiabschnitt, an dem die Dateiverarbeitung beginnen soll.

Wird „pfadname“ mit OPEN EXTEND eröffnet/erweitert, gibt VSEQ den Dateiabschnitt an, an dem die Erweiterung beginnt.

Im Zusammenhang mit OPEN REVERSE können einzelne Bänder einer Datei verarbeitet werden, automatischer Bandwechsel ist jedoch nicht möglich.

= (L=list-poss (255): <integer 1..255>)
gibt an, in welcher Reihenfolge die Dateiabschnitte verarbeitet werden sollen. Diese Angabe ist nur zulässig für Eingabedateien, nicht für Ausgabedateien. Für Dateien, die mit OPEN REVERSE eröffnet werden, kann nur eine Dateiabschnittsnummer angegeben werden; automatischer Bandwechsel wird nicht unterstüzt.

WORKFIL
Nur ab VERSION=3, für Dateien auf SM-Pubsets:
legt fest, ob die Datei auf einem Volume-Set für Arbeitsdateien oder auf einem Volume-Set mit permanenter Datenhaltung angelegt wird. Volume-Sets für Arbeitsdateien werden zu einem von der Systemverwaltung festgelegten Zeitpunkt wieder gelöscht. Arbeitsdateien können nicht auf einem Net-Storage-Volume angelegt werden. Wird die Datei mittels nichtphysikalischer Allokierung auf einem SM-Pubset angelegt und ist WORKFIL nicht angegeben, so wird die Datei auf einem Volume-Set mit permanenter Datenhaltung abgelegt. Die Angabe von WORKFIL wird in folgenden Fällen abgewiesen:
– für Generationen
– wenn die Datei bereits Speicherplatz belegt
– wenn SPACE mit nichtpositiver Primärzuweisung angegeben ist
– wenn die Datei auf einer Privatplatte zu liegen kommt.

= NO
Die Datei wird auf einem Volume-Set mit permanenter Datenhaltung abgelegt.

= YES
Die Datei wird auf einem Volume-Set für Arbeitsdateien angelegt. Die Angabe ist nicht erlaubt für temporäre Dateien.

WRCHK
Für die Verarbeitung von Plattendateien:
gibt an, ob bei Schreiboperationen „Kontroll-Lesen“ erfolgt. „WRCHK“ wird nicht in den Katalogeintrag aufgenommen und muss daher vor jeder Verarbeitung oder Dateieröffnung wiederholt werden.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

Kontroll-Lesen: Prüfung auf Aufzeichnungsfehler (→ Error-Recovery-Maßnahmen). Ist der Fehler nicht behebbar, wird der EXLST-Ausgang ERRADR angestoßen. Das Kontroll-Lesen geht zulasten der Performance.

= NO
Es erfolgt kein Kontroll-Lesen.

= YES
Es erfolgt Kontroll-Lesen.

WROUT
Für ISAM-Verarbeitung:
WROUT steuert das Zurückschreiben geänderter Blöcke auf die Platte. Bei Shared-Update-Verarbeitung, in taskübergreifenden ISAM-Pools und in tasklokalen ISAM-Pools, für die WRITE-IMMEDIATE eingestellt wurde, gilt implizit WROUT=YES: geänderte Blöcke werden stets sofort auf Platte zurückgeschrieben.

Ist weder im FILE-Makro noch im FCB-Makro ein Wert angegeben, so wird bei Dateieröffnung die Voreinstellung des FCB-Makros wirksam.

= NO
Der geänderte Block wird erst dann zurückgeschrieben, wenn der Inhalt des betreffenden Pufferbereichs ersetzt werden muss, spätestens beim Schließen der Datei.

= YES
Jeder geänderte Block wird sofort zurückgeschrieben, sodass die Konsistenz der Daten auf Platte und im virtuellen Speicher stets gewährleistet ist. Allerdings ist damit auch eine erhöhte Ein-/Ausgaberate verbunden.

Hinweise zur Programmierung

Aufbau der Operandenliste bei Angabe von VERSION=0 oder VERSION=1

Die Operandenliste des FILE-Makros besteht aus mehreren festen und variablen Bereichen:

```
fester Bereich 1
(DSECT kann mit Makro IDPFL erzeugt werden; siehe unten)

variabler Bereich für VOLUME- bzw. TVSN-Angaben
(falls der Operand VOLUME / TVSN in Listenform angegeben
wird)
                              :
                              :

variabler Bereich für MOUNT-Angaben
(falls der Operand MOUNT in Listenform angegeben wird)
                              :
                              :

fester Bereich 2 (FILE-Erweiterung)
(DSECT kann mit Makro IDPFX erzeugt werden; siehe unten)

variabler Bereich für DVOLUME-Angaben
falls der Operand DVOLUME in Listenform angegeben wird)
                              :
                              :

variabler Bereich für VSEQ-Angaben
(falls der Operand VSEQ in Listenform angegeben wird)
                              :
                              :
```

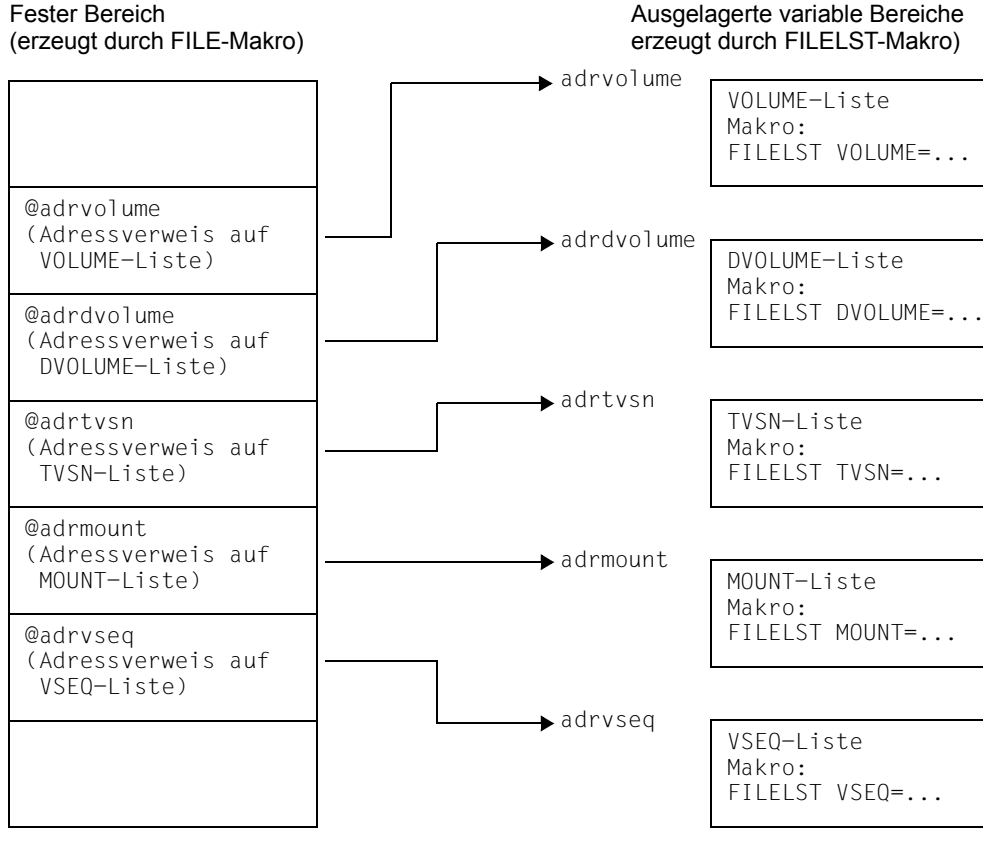
Für die beiden festen Bereiche der Operandenliste werden mit MF=D im FILE-Makroaufruf zwei DSECTs erzeugt. Der Makroaufruf IDPFL erzeugt eine DSECT für den festen Bereich 1, der Makroaufruf IDPFX eine DSECT für die FILE-Erweiterung. Die DSECTS aus IDPFL und IDPFX unterstützen jedoch nur das bereits vor der BS2000-Version V9.5 vorhandene Makroaufrufformat.

Aufbau der Operandenliste ab VERSION=2

Dieses Format unterstützt alle ab BS2000 V10.0 neu eingeführten Operanden und Operandenwerte.

Im Unterschied zur Operandenliste bei VERSION=0/1 kann der Anwender ab VERSION=2 die variablen Teile für die VOLUME-, TVSN-, MOUNT-, DVOLUME- und VSEQ-Angaben in eigene Bereiche außerhalb der Operandenliste auslagern, indem er bei diesen Operanden jeweils mit der Angabe „@adr“ auf eine symbolische Adresse „adr“ im Programm verweist, an der er mit dem Makro FILELST eine Liste mit den zugehörigen Operandenwerten angelegt hat.

Die vom FILE-Makro erzeugte Operandenliste ist in diesem Fall ein Bereich fester Länge, der lediglich Adressverweise auf die ausgelagerten variablen Listen enthält. Sie hat folgenden Aufbau:



DSECTs für den festen Bereich und die ausgelagerten variablen Bereiche können mit der D-Form des FILE- und des FILELST-Makros erzeugt werden.

Beispiel

Das Programm TAPEFIL liest die Banddatei TAPE.FILE über den Dateikettungsnamen INTAPE. Die Liste der benötigten Datenträger wird mit dem Operanden TVSN im FILE-Makro spezifiziert:

```
TAPEFIL  START
         .
         .
FILE  TAPE.FILE,LINK=INTAPE,TVSN=@TVSNLIST,VERSION=3,MF=L ————————————  (1)
         .
         .
TVSNLIST FILELST TVSN=(VOL003,VOL009,VOL017) ——————————————————————————  (2)

         .
         .
         END
```

(1) Die Angabe @TVSNLIST im Operanden TVSN erzeugt in der Operandenliste des FILE-Makros einen Adressverweis auf die symbolische Adresse TVSNLIST. An dieser Stelle erwartet der FILE-Makro den (variablen) Bereich mit der Liste der TVSN-Werte.

(2) Der Makro FILELST erzeugt an der Adresse TVSNLIST eine Liste mit den Werten für den Operanden TVSN im FILE-Makro.

Hinweise für die Verarbeitung von Banddateien

Operand STATE = FOREIGN

Im Katalogeintrag wird ein FOREIGN-Kennzeichen gesetzt, sodass die Eigenschaften der Datei nicht mit einem CATAL-Makroaufruf geändert werden können. Das FOREIGN-Kennzeichen wird bei der Dateieröffnung zurückgesetzt.

Gehören aufeinander folgende Dateigenerationen einer Gruppe zum gleichen MF/MV-Set, sollte auf keinen Fall im CATAL-Makroaufruf DISP=REUSE angegeben werden, da dies zur Zerstörung von Dateigenerationen führen kann.

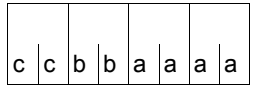
Die Vorgehensweise bei der Übernahme einer FOREIGN-Banddatei entspricht nicht der bei privaten Plattendateien. Dies liegt daran, dass der Katalogeintrag einer FOREIGN-Plattendatei eindeutig ist. Für FOREIGN-Banddateien könnte diese Eindeutigkeit dann erzielt werden, wenn die Benutzerkennungen der Datei-Eigentümer in dem System existieren, in das die Datei übernommen werden soll. Ist jedoch die Benutzerkennung nicht vorhanden, ist es nicht möglich, die Eigentümerkennung auf dem Band zu ändern (HW-Einschränkung führt zur Zerstörung der Datei). Auch wenn der Systemverwalter eine Datei für eine schon bestehende Benutzerkennung übernimmt, ist das keine Garantie für die Einmaligkeit des Katalogeintrages, da er die Datei auch unter einer weiteren Benutzerkennung katalogisieren kann.

Dennoch sind Banddateien mit Standardkennsätzen genauso geschützt gegen Widersprüchlichkeiten zwischen Dateieigenschaften der Kennsätze und der des Katalogeintrages wie Plattendateien durch die Einschränkungen im CATAL-Makroaufruf. Lediglich die Möglichkeit des Dateieigentümers, die Dateieigenschaften mittels Angabe des SECLEV= LOW im FCB zu ändern, birgt in sich einen Unsicherheitsfaktor. Deshalb sollten für eine Datei in einem System dann nicht mehrere Katalogeinträge vorhanden sein, wenn auch der Eigentümer dieser Datei in dem gleichen System arbeitet.

Returncodes

Der Fehlercode wird ab Version 3 nur noch im Standardheader der Parameterliste und nicht mehr wie in Version 2 im Mehrzweckregister 15 zurückgeliefert. Der Standardheader darf nicht im Read-only-Bereich liegen, sonst erfolgt Programmterminierung.

Standard-header:



Über die Ausführung des Makros FILE wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Kein Fehler
	X'40'	X'0501'	Angeforderter Katalog nicht verfügbar
	X'82'	X'0502'	Angeforderter Katalog im Ruhezustand
	X'40'	X'0503'	Falsche Information im MRSCAT
	X'82'	X'0504'	Fehler beim Katalogzugriff
	X'40'	X'0505'	Fehler bei der Rechner-Kommunikation (MRS)
	X'80'	X'0506'	Operation wegen Masterwechsel abgebrochen
	X'40'	X'0510'	Fehler beim Aufruf einer internen Funktion
	X'40'	X'0511'	Keine Allokierung wegen MVDF-Inkonsistenz
	X'40'	X'0512'	Katalogkennung ist nicht im MRSCAT eingetragen
	X'40'	X'0515'	Aufruf wurde von System-Exit-Routine abgewiesen
	X'40'	X'051B'	Benutzerkennung im angegebenen Pubset unbekannt
	X'40'	X'051C'	Kein Zugriffsrecht auf angegebenen Pubset
	X'40'	X'051D'	LOGON-Passwort auf angegebenem Pubset anders
X'02'	X'00'	X'051E'	Nur Teilallokierung wegen MVDF-Inkonsistenz
	X'20'	X'0531'	Unerwarteter Fehler beim Katalogzugriff
	X'82'	X'0532'	Datei gesperrt weil in Gebrauch
	X'40'	X'0533'	Datei nicht gefunden
	X'82'	X'0534'	Privater Datenträger kann nicht zugewiesen werden
	X'40'	X'0535'	Keine Zugriffsberechtigung auf den Katalogeintrag der Datei
	X'20'	X'0536'	Fehler im Dateiverwaltungssystem
	X'40'	X'053A'	Fehler beim Ändern des F1-Labels auf privater Platte
	X'20'	X'053B'	Systemfehler beim Katalogzugriff
	X'82'	X'053C'	Katalog-Datei des Pubsets ist voll
	X'40'	X'053D'	Katalog oder F1-Etikett-Block ist zerstört

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'053E'	Datei auf privatem Datenträger bereits katalogisiert
	X'82'	X'053F'	Datei ist von einer anderen Task reserviert
	X'40'	X'0540'	Pubset enthält kein passendes Volume-Set
	X'82'	X'0541'	Kein Plattenspeicherplatz zugewiesen
	X'82'	X'0542'	Gerät nicht verfügbar / Platte exklusiv
	X'20'	X'0543'	Fehlerhafter Allokator-Parameterbereich
	X'20'	X'0544'	Falsch formatierter Katalogeintrag
	X'40'	X'0545'	Public Volume nicht angeschlossen
X'02'	X'00'	X'0546'	Katalogeintrag der Datei ist voll
	X'40'	X'0547'	Datenträger kann nicht montiert werden
	X'82'	X'0548'	Speichermangel
	X'20'	X'0549'	Systemfehler bei REQM- oder AQIR-Aufruf
X'02'	X'00'	X'054A'	Speicherplatz nur teilweise zugewiesen
	X'40'	X'054B'	Zur angegebenen Catid ist kein Volume-Set verfügbar
	X'82'	X'054D'	Speicherplatz-Kontingent erschöpft
	X'82'	X'0550'	Datei geöffnet und damit gesperrt
	X'01'	X'0551'	VSN für Banddatei mehrfach angegeben
	X'01'	X'0553'	Unzulässige absolute Speicherplatzanforderung
	X'01'	X'0554'	Format des Dateinamens unzulässig
	X'40'	X'0555'	STATE=FOREIGN: Datei bereits katalogisiert
	X'01'	X'0556'	STATE=FOREIGN: Gerätetyp ungültig oder fehlend
	X'40'	X'0557'	Fehlerhafte VSN-Angabe
	X'01'	X'0558'	Public-VSN unzulässig
	X'01'	X'0559'	Unzulässige Angabe bei MOUNT
	X'82'	X'055A'	Bandgeräte zurzeit belegt
	X'40'	X'055C'	F1-Label fehlt
	X'40'	X'055D'	Benutzer hat kein Recht zur physikalischen Allokierung
	X'40'	X'055E'	Fremde Benutzerkennung für nicht katalogisierte Datei
	X'40'	X'055F'	Datenträger konnte nicht belegt werden
	X'01'	X'0576'	Fehlerhafte Operandenkombination oder nicht gelöschte UNUSED-Felder
	X'20'	X'0578'	Interner Fehler bei der Überprüfung der Zugriffsrechte
	X'01'	X'0579'	Ungültiger Operand für temporäre Datei oder für Arbeitsdatei

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'057A'	Storage-Klasse unverträglich mit Dateieigenschaften
	X'40'	X'057B'	Unterlaubter Operand für migrierte Datei
	X'40'	X'057C'	HSMS hat Recall abgewiesen
	X'40'	X'057E'	HSMS nicht verfügbar
	X'01'	X'0590'	Gerätetypangabe für privaten Datenträger fehlt
	X'01'	X'0592'	Privatplattendatei hat Katalogeintrag ohne Extents oder Gerätetyp-bestimmung für Public-Platte abgewiesen
	X'01'	X'0593'	Absolutzuweisung: Anzahl von Halbseiten unzulässig
	X'82'	X'0594'	Nicht genug virtueller Speicher verfügbar
	X'01'	X'0595'	Unzulässiger Mix von Public- und Private-VSNs
	X'01'	X'0596'	Gerätetypangabe nicht gemäß Katalogeintrag
	X'01'	X'0597'	Absolutzuweisung: erste Halbseite nicht auf Unitgrenze
	X'01'	X'0599'	Operand wird in der Remote-Version nicht unterstützt
	X'01'	X'05A3'	Fehlerhafte SPACE-Angabe
	X'01'	X'05A4'	Fehlerhafte Verwendung von DSPACE/DVOLUME/DDEVICE
	X'01'	X'05A8'	Gerätetyp im System nicht vorhanden
	X'82'	X'05B0'	Kein passendes Bandgerät verfügbar
	X'82'	X'05B1'	Für die Datei besteht eine Dateisperre
	X'40'	X'05B4'	Datenträgeranforderung wurde abgewiesen
	X'40'	X'05BD'	Unzulässige Kombination von Datei- und Volume-Set-Eigenschaften
	X'01'	X'05C2'	Kettungsname = X'0000000000000000'
	X'82'	X'05C3'	Zu löschende Dateigeneration gesperrt
	X'40'	X'05C4'	Bei Operatormeldung trat ein Fehler auf
	X'20'	X'05C7'	Interner Fehler im DMS
	X'82'	X'05C8'	CE-Limit für Benutzerkennung überschritten
	X'20'	X'05CA'	Interner Fehler bei Änderung des CE-Limits
	X'40'	X'05D8'	Datei mit Kennwort geschützt
	X'40'	X'05DA'	Speicherplatzfreigabe auf fremder Benutzerkennung
	X'01'	X'05DF'	Unzulässige Angabe zu BLIM / CHKPT
	X'20'	X'05E0'	Dateisperre wegen Systemfehler bei Speicherplatzverwaltung
	X'01'	X'05E8'	Dateiname ungültig für Plattendatei
	X'01'	X'05EE'	Pfadname nach Komplettierung zu lang
	X'01'	X'05EF'	Dateischutz mit ACL/GUARD geht nur für Public-Dateien

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'01'	X'05FA'	Pubset nicht lokal zugreifbar
	X'40'	X'05FC'	Benutzerkennung nicht eingetragen
	X'40'	X'05FD'	Datei durch Freigabedatum oder Zugriffsart geschützt
	X'40'	X'0606'	Datenträger-Anforderung von MAREN abgewiesen
	X'40'	X'0609'	Speicherplatzfreigabe für Systemdatei nicht erlaubt
	X'40'	X'060D'	Fehlerhafter Dateiname für Referenzdatei angegeben
	X'40'	X'060E'	Referenzdatei nicht gefunden oder nicht zugreifbar
	X'40'	X'0613'	Fehlerhafte Angabe einer Storage-Klasse
	X'40'	X'0640'	Zugriff auf Net-Storage wird vom Subsystem ONETSTOR wegen Kommunikationsproblemen mit dem Net-Client abgewiesen
	X'40'	X'0641'	Datei auf Net-Storage bereits vorhanden
	X'40'	X'0642'	Große Dateien auf dem angegebenen Pubset nicht erlaubt
	X'40'	X'0643'	Net-Client meldet Zugriffsfehler
	X'40'	X'0644'	Net-Client meldet internen Fehler
	X'40'	X'0645'	Datei auf Net-Storage nicht vorhanden
	X'40'	X'0647'	Angegebener Dateityp stimmt nicht mit dem Katalog-Eintrag der Datei überein
	X'40'	X'0648'	Angabe von Dateityp, Device und Volume passen nicht zusammen
	X'40'	X'0649'	Net-Server meldet POSIX-ACL-Fehler
	X'40'	X'064A'	Net-Client meldet, dass Zugriff auf Dateien auf dem Net-Storage-Volume verboten ist
	X'40'	X'064B'	Zugriff auf Node-Files vom Net-Client nicht unterstützt
	X'40'	X'0652'	Absolute Speicherplatz Anforderung auf Net-Storage nicht erlaubt
	X'40'	X'0666'	Datei ist gegen geforderten Zugriff geschützt
	X'40'	X'0683'	Datei existiert bereits
	X'40'	X'0689'	Operand nur erlaubt für Datei ohne Speicher
	X'40'	X'06B5'	Datei ist nicht ordnungsgemäß geschlossen
	X'01'	X'06C7'	Ungültige Generationsnummer
	X'01'	X'06C8'	Attribut unzulässig für Dateigenerationen
	X'40'	X'06CD'	FGG gegen Erweitern geschützt
	X'01'	X'06CF'	Angabe einer FGG unzulässig
	X'40'	X'06D0'	STATE=FOREIGN für nicht existierende Dateigeneration
	X'40'	X'06D1'	FGG-Index durch andere Task gesperrt
	X'01'	X'06DA'	Unzulässiger Public-Private-Mix für FGG

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'01'	X'06DF'	Unzulässige Angabe zu FSEQ/VSEQ/TSET
	X'01'	X'06FD'	Ungültige Adresse des Parameterbereiches
	X'40'	X'06FF'	BCAM-Verbindung abgebrochen
	X'01'	X'FFFF'	Falsche Funktionsnummer im Standardheader
	X'03'	X'FFFF'	Falsche Versionsnummer im Standardheader

Versionsunterschiede VERSION=0/1/2/3

Operand	Vers=0	Vers=1	Vers=2	Vers=3	Bemerkungen zu Operandenwerten
MF=E	x	x	x	x	
VERSION	x	x	x	x	
MF=C	-	x	x	x	
PREFIX	-	x	x	x	
VERSION	-	x	x	x	
MF=D	-	x	x	x	
PREFIX	-	x	x	x	
VERSION	-	x	x	x	
MF=L	x	x	x	x	
*DUMMY	x	x	x	x	
pfadname	x	x	x	x	
AVAIL	-	-	-	x	
BLIM	x	x	x	x	
BLKCTRL	-	x	x	x	DATA2K und DATA4K erst ab Vers=2 *BY-PROG erst ab Vers=3
BLKSIZE	x	x	x	x	*BY-PROG erst ab Vers=3
BUFOFF	x	x	x	x	*BY-PROG erst ab Vers=3
BYPASS	x	x	x	x	
CHAINIO	x	x	x	x	
CHKPT	x	x	x	x	
CLOSE	-	-	x	x	KEEP-DATA-IN-CACHE erst ab Vers=3
CLOSMSG	-	x	x	x	
CODE	x	x	x	x	*BY-PROG erst ab Vers=3 ISO7D erst ab Vers=3

Operand	Vers=0	Vers=1	Vers=2	Vers=3	Bemerkungen zu Operandenwerten
MF=L (Fortsetzung)					
DATATTR	-	-	-	x	
DDEVICE	x	x	x	x	
DESTOC	-	x	x	x	
DEVICE	x	x	x	x	
DISKWR	-	-	-	x	
DSPACE	x	x	x	x	
DUPEKY	x	x	x	x	
DVOLUME	x	x	x	x	@adr erst ab Vers=2
EXC32GB	-	-	-	x	
FCBTYPE	x	x	x	x	*BY-PROG erst ab Vers=3
FSEQ	x	x	x	x	
IOPERF	-	-	x	x	
IOUSAGE	-	-	x	x	
KEYLEN	x	x	x	x	*BY-PROG erst ab Vers=3
KEYPOS	x	x	x	x	*BY-PROG erst ab Vers=3
LABEL	x	x	x	x	*BY-PROG erst ab Vers=3
LINK	x	x	x	x	
LOCKENV	-	-	-	x	
LOGLEN	x	x	x	x	*BY-PROG erst ab Vers=3
MOUNT	x	x	x	x	@adr erst ab Vers=2
NFTYPE	-	-	-	x	
OPEN	x	x	x	x	
OVERLAP	x	x	x	x	
PAD	x	x	x	x	
POOLLNK	-	x	x	x	
POOLSIZ	-	-	-	x	
RECFORM	x	x	x	x	*BY-PROG erst ab Vers=3
RECSIZE	x	x	x	x	*BY-PROG erst ab Vers=3
RETPD	x	x	x	x	
SECLEV	x	x	x	x	
SHARUPD	x	x	x	x	WEAK erst ab Vers=2
SPACE	x	x	x	x	*KEEP erst ab Vers=2

Operand	Vers=0	Vers=1	Vers=2	Vers=3	Bemerkungen zu Operandenwerten
MF=L (Fortsetzung)					
STATE	x	x	x	x	
STOCLAS	-	-	-	x	
STREAM	x	x	x	x	
TAPEWR	-	x	x	x	
TPMARK	x	x	x	x	
TRANS	x	x	x	x	
TSET	x	x	x	x	
TVSN	x	x	x	x	@adr erst ab Vers=2
VALLEN	x	x	x	x	*BY-PROG erst ab Vers=3
VALPROP	x	x	x	x	*BY-PROG erst ab Vers=3
VOLSET	-	-	-	x	
VOLUME	x	x	x	x	@adr erst ab Vers=2 REMOVE-UNUSED erst ab Vers=3
VSEQ	x	x	x	x	@adr erst ab Vers=2
WORKFIL	-	-	-	x	
WRCHK	x	x	x	x	
WROUT	x	x	x	x	

Legende

x Operand ist in der Makroversion verfügbar
\- Operand ist in der Makroversion nicht verfügbar
Vers Version

In der Tabelle sind unter MF=L die Stellungsoperanden vor den Schlüsselwortoperanden eingeordnet.

FILELST – Variable Operandenbereiche für FILE-Makro erzeugen

Makrotyp: S-Typ (L-Form/D-Form/C-Form) (siehe Seite 870)

Der Makro FILELST erzeugt die separaten Listen für die Operanden VOLUME, DVOLUME, TVSN, MOUNT und VSEQ des FILE-Makros. Im FILE-Makro können diese Listen durch die Angabe des Wertes @adr bei diesen Operanden angesprochen werden, wobei adr die symbolische Adresse mit dem FILELST-Aufruf bezeichnet.

Formate

Format 1: L-Form

Operation	Operanden
FILELST	[MF=L] [VOLUME={vsn | (vsn,...)}] [,DVOLUME={vsn | (vsn,...)}] [,TVSN={vsn | (vsn,...)}] [,MOUNT={zahl | (zahl,...)}] [,VSEQ={zahl | (zahl,...)}]

Operandenbeschreibung

VOLUME
Es kann eine Archivnummer oder eine Liste von maximal 255 Archivnummern angegeben werden.
Es gilt die Beschreibung des VOLUME-Operanden im FILE-Makro (siehe Seite 508). Die Angabe VOLUME=PRIVATE oder VOLUME=(PRIVATE,n) ist jedoch nur im FILE-Makro zulässig.

DVOLUME
Es kann eine Archivnummer oder eine Liste von maximal 255 Archivnummern angegeben werden.
Es gilt die Beschreibung des DVOLUME-Operanden im FILE-Makro (siehe Seite 481). Die Angabe DVOLUME=PRIVATE oder DVOLUME=(PRIVATE,n) ist jedoch nur im FILE-Makro zulässig.

MOUNT
Es können maximal 255 Datenträger angefordert werden.

Es gilt die Beschreibung des MOUNT-Operanden im FILE-Makro (siehe Seite 488). Die Angabe MOUNT=0 ist jedoch nur im FILE-Makro zulässig.

TVSN
Es kann eine Archivnummer oder eine Liste von maximal 255 Archivnummern angegeben werden.
Es gilt die Beschreibung des TVSN-Operanden im FILE-Makro (siehe Seite 506).

VSEQ
Es können maximal 255 Dateiabschnitte angegeben werden.

Es gilt die Beschreibung des VSEQ-Operanden im FILE-Makro (siehe Seite 510). Es wird jedoch nur die Listenangabe unterstützt, die Einzelangabe ist nur im FILE-Makro zulässig.

Format 2: D-Form/C-Form

Operation	Operanden
FILELST	MF={D | C}[,PREFIX={I | pre}][,MACID={DBL | macid}] [,LIST={VOLUME | (VOLUME,zahl) | DVOLUME | (DVOLUME,zahl) | TVSN | (TVSN,zahl) | MOUNT | (MOUNT,zahl) | VSEQ | (VSEQ,zahl)}]

Operandenbeschreibung

LIST
Der Operand legt fest, für welche der von FILELST erzeugten Listen eine CSECT bzw. DSECT generiert wird.

= VOLUME
= (VOLUME,zahl)
Es wird eine CSECT bzw. DSECT für die VOLUME-Liste erzeugt. Spezifiziert der Anwender den Operandenwert in der Form (VOLUME,zahl) – die runden Klammern sind Bestandteil des Wertes und müssen mit angegeben werden – , so kann er mit zahl angeben, wie viele Elemente die VOLUME-Liste enthält, für die die CSECT bzw. DSECT erzeugt werden soll.

= DVOLUME
= (DVOLUME,zahl)
Es wird eine CSECT bzw. DSECT für die DVOLUME-Liste erzeugt. Spezifiziert der Anwender den Operandenwert in der Form (DVOLUME,zahl) – die runden Klammern sind Bestandteil des Wertes und müssen mit angegeben werden – , so kann er mit zahl angeben, wie viele Elemente die DVOLUME-Liste enthält, für die die CSECT bzw. DSECT erzeugt werden soll.

= TVSN
= (TVSN,zahl)
Es wird eine CSECT bzw. DSECT für die TVSN-Liste erzeugt. Spezifiziert der Anwender den Operandenwert in der Form (TVSN,zahl) – die runden Klammern sind Bestandteil des Wertes und müssen mit angegeben werden – , so kann er mit zahl angeben, wie viele Elemente die TVSN-Liste enthält, für die die CSECT bzw. DSECT erzeugt werden soll.

= MOUNT
= (MOUNT,zahl)
Es wird eine CSECT bzw. DSECT für die MOUNT-Liste erzeugt. Spezifiziert der Anwender den Operandenwert in der Form (MOUNT,zahl) – die runden Klammern sind Bestandteil des Wertes und müssen mit angegeben werden – , so kann er mit zahl angeben, wie viele Elemente die MOUNT-Liste enthält, für die die CSECT bzw. DSECT erzeugt werden soll.

= VSEQ
= (VSEQ,zahl)
Es wird eine CSECT bzw. DSECT für die VSEQ-Liste erzeugt. Spezifiziert der Anwender den Operandenwert in der Form (VSEQ,zahl) – die runden Klammern sind Bestandteil des Wertes und müssen mit angegeben werden – , so kann er mit zahl angeben, wie viele Elemente die VSEQ-Liste enthält, für die die CSECT bzw. DSECT erzeugt werden soll.

PREFIX
legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: PREFIX = I

= pre
„pre“ ist ein ein Zeichen langes Präfix, mit dem die generierten Feldnamen und Equates beginnen sollen.

MACID
legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: MACID = DBL

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

FPAMACC – FASTPAM-Dateizugriffe formulieren

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Der FPAMACC Makro realisiert die Funktion ACCESS FILE. Es können auf die über die OPENID referenzierte Datei Zugriffe formuliert werden.

Format

Operation	Operanden
FPAMACC	[,OPENID={zahl | adr | (r)}] [,LEN={länge | adr | (r)}] [,BLOCK={zahl | adr | (r)}] [,IOAREA={adr | (r)}] [,OPCODE={*READ | *WRITE | *READ_WAIT | *WRITE_WAIT | *READ_EQUALIZE | *WAIT | adr | (r)}]

(Teil 1 von 2)

Operation	Operanden
	[,WAITLST={adr / (r)}] [,CHAIN={adr / (r)}] [,POSTCD={zahl / adr / (r)}] MF=L
	MF=E,PARAM={adr / (r)}
	MF=D[,PREFIX={F / pre}]
	MF={C / M}[,PREFIX={F / pre}][,MACID={ACC / macid}]

(Teil 2 von 2)

Operandenbeschreibung

BLOCK
Bestimmt die Nummer des ersten zu übertragenden logischen FASTPAM-Blockes innerhalb der Datei.
Die Blockgröße wurde beim Makro FPAMSRV, Funktion OPEN, Operand BLKSIZE bestimmt. Es sind nur ganzzahlige Werte zugelassen.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= zahl
Ist die direkte Angabe eines dezimalen numerischen Wertes für die Nummer des ersten zu übertragenden logischen Blockes. Der Wert wird begrenzt durch die maximale Größe einer Datei in 4-KB-Seiten minus 1:

$1 \leq$ zahl $\leq$ 8388606 bei LARGE_FILE=*FORBIDDEN (siehe Makro FPAMSRV)

$1 \leq$ zahl $\leq$ 1073741823 bei LARGE_FILE=*ALLOWED (siehe Makro FPAMSRV)

= adr
Ist die symbolische Adresse eines 4 Byte langen Feldes, das den numerischen Wert enthält (binär).

= (r)
Ist ein Register, das den numerischen Wert enthält.

CHAIN
Spezifiziert die Anfangsadresse einer anderen FPAMACC-Parameterliste, deren Auftrag an den der aktuellen Parameterliste gekettet werden soll. Auf diese Weise können bis zu 5000 Parameterlisten miteinander verbunden werden. Alle auf diese Weise miteinander verketteten Aufträge werden innerhalb eines SVC angenommen bzw. ausgeführt.
Die Aufträge werden in der Reihenfolge ihrer Verkettung ausgeführt. Wird in einem der Aufträge ein Parameterfehler entdeckt, werden auch alle anderen mit Returncode „CHAIN ERROR“ abgewiesen. Bei jedem anderen Fehler wird ab dem Zeitpunkt, da er erkannt wird, kein weiterer Auftrag mehr bearbeitet außer beim Returncode FACCPNAC. Alle nachfolgenden Aufträge werden ebenfalls abgewiesen. Bei asynchronen Ein-Ausgaben mit dazu geketteten Wait-Parameterlisten muss deshalb der Anwender bei einem Fehler abgewiesene Wait-Operationen wiederholen.
Tritt ein Fehler erst auf, nachdem alle Aufträge erfolgreich angestartet worden sind (z.B. ein Ein-Ausgabefehler in einer Kette asynchroner Read/Write-Operationen mit Eventing), werden alle anderen Aufträge unabhängig von dem fehlerhaften behandelt. Jede Parameterliste muss also für sich ausgewertet werden.
Vergleiche dazu auch den Abschnitt „Fehlerbehandlung bei Parameterlistenkettung“ auf Seite 537.

Bei der Form MF=L ist nur die symbolische Adresse erlaubt, wobei symbolische Namen innerhalb einer DSECT ausgeschlossen sind, da ihre Adresse erst zur Laufzeit bekannt ist.

Hinweis

Bei Eventing wird für jeden einzelnen Auftrag ein Signal gesendet.

= adr
Ist die symbolische Adresse (Name) der nächsten zu bearbeitenden Parameterliste.

= (r)
Ist ein Register, das die Anfangsadresse der nächsten zu bearbeitenden Parameterliste enthält.

IOAREA
Spezifiziert die auf 4 KByte ausgerichtete Anfangsadresse des Ein-/Ausgabepuffers, der innerhalb des beim OPEN angegebenen IO-Area-Pools liegen muss.
Bei Datenräumen wird automatisch der beim ENABLE IOAREA POOL angegebene ALET benutzt.

Bei der Form MF=L ist nur die symbolische Adresse erlaubt, wobei symbolische Namen innerhalb einer DSECT ausgeschlossen sind, da ihre Adresse erst zur Laufzeit bekannt ist.

= adr
Ist die symbolische Adresse (Name) des Bereichs.

= (r)
Ist ein Register, das die Anfangsadresse des Ein-/Ausgabepuffers enthält.

LEN
Bestimmt die zu übertragende Datenlänge in logischen Blöcken. Die Blockgröße wird im Makro FPAMSRV, Funktion OPEN, Operand BLKSIZE festgelegt. Es sind nur ganzzahlige Werte zwischen 1 und 8 zugelassen, die den beim ENABLE ENVIRONMENT angegebenen MAXIOLN-Wert nicht überschreiten.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= länge
Direkte Angabe eines dezimalen numerischen Wertes.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den numerischen Wert enthält (binär).

= (r)
Ist ein Register, das den numerischen Wert enthält.

MACID
Legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

= ACC
Voreinstellung: MACID=ACC

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang, Seite 870 beschrieben.

OPCODE
Kennzeichnet die Art des Auftrags.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *READ
Asynchrones Lesen von logischen Blöcken. Im Subcode2 wird hinterlegt, ob der Auftrag synchron oder asynchron erledigt wurde. Wurde beim OPEN, Operand EVENTNG=*NO angegeben und ist der Auftrag nicht synchron beendet worden, muss er mit OPCODE=*WAIT abgeschlossen werden, sonst werden alle folgenden Aufträge mit dieser FPAMACC-Parameterliste abgewiesen.

= *WRITE
Asynchrones Schreiben von logischen Blöcken. Im Subcode2 wird hinterlegt, ob der Auftrag synchron oder asynchron erledigt wurde. Wurde beim OPEN EVENTNG=*NO angegeben und ist der Auftrag nicht synchron beendet worden, muss er mit OPCODE=*WAIT abgeschlossen werden, sonst werden alle folgenden Aufträge mit dieser FPAMACC-Parameterliste abgewiesen.

= *READ_WAIT
Synchrones Lesen von logischen Blöcken.

= *WRITE_WAIT
Synchrones Schreiben von logischen Blöcken.

= *READ_EQUALIZE
Synchrones Lesen von logischen Blöcken mit gleichzeitiger Egalisierung der DRV-Platten im angegebenen Bereich innerhalb der Datei. Im nicht-DRV-Betrieb wirkt *READ_EQUALIZE wie *READ_WAIT (zu DRV siehe auch Handbuch „DRV“ [15]).

= *WAIT
Warten auf das Ende eines asynchronen Auftrags (*READ oder *WRITE). Diese Operation darf nur von der Task ausgeführt werden, die den Auftrag gestartet hat.
Mit dem Operanden WAITLST wird die Adresse der FPAMACC-Parameterliste angegeben, auf deren Ein-/Ausgabe gewartet werden soll. Dies kann auch dieselbe Parameterliste sein, muss aber im selben Environment liegen und dieselbe OPENID enthalten wie die WAIT-Parameterliste.

Die Angabe *WAIT ist in folgenden Fällen unzulässig:
– Nach den synchronen Operationen *READ_WAIT, *WRITE_WAIT und *READ_EQUALIZE.
– Wenn schon ein *WAIT erfolgt ist.
– Wenn der asynchrone Auftrag synchron beendet wurde (Subcode2=FACCSYTE nach *READ/*WRITE).
– Wenn beim OPEN EVENTNG=*YES angegeben wurde.

= adr
Ist die symbolische Addresse eines 1 Byte langen Feldes, das den Wert für OPCODE enthält.

= (r)
Ist ein Register, das den Wert für OPCODE enthält.

OPENID
Bezeichnet die Kurzkennung des OPEN, für den die FPAMACC Operation ausgeführt werden soll.
Sie muss nach erfolgreicher OPEN Verarbeitung aus der FPAMSRV- in die FPAMACC-Parameterliste übertragen werden.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= zahl
Ist direkte Angabe eines dezimalen numerischen Wertes für die Kurzkennung des OPEN.

= adr
Ist die Adresse eines 4 Byte langen Feldes, das die Kurzkennung enthält.

= (r)
Ist ein Register, das die Kurzkennung enthält.

PARAM
Bezeichnet die Adresse der Operandenliste. Der Operand wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

POSTCD
Enthält die Daten, die beim Börsensignal mitgegeben werden. Dieser Parameter wird nur bei EVENTNG=*YES ausgewertet.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= adr
Ist die Adresse eines 2 Byte langen Feldes, das den POSTCD enthält.

= (r)
Ist ein Register, das den POSTCD enthält (unterste 2 Byte).

PREFIX
Legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

= F
Voreinstellung: PREFIX=F

= pre
„pre“ ist ein ein Zeichen langes Präfix, mit dem die generierten Feldnamen und Equates beginnen sollen.

WAITLST
Spezifiziert die Anfangsadresse der FPAMACC-Parameterliste, auf deren Ein-Ausgabe mit der Operation WAIT gewartet werden soll. Dies kann auch dieselbe Parameterliste sein, muss aber im selben Environment liegen und dieselbe OPENID enthalten.

Ein Fehler wird in der Parameterliste gemeldet, mit der die fehlerhafte Operation angestartet wurde. Bei einem Ein-/Ausgabefehler z.B. erhält die Wait-Parameterliste den Returncode „SUCCESSFUL_PROCESSING“ und die Read/Write-Parameterliste „IO_ERROR“. WAITLST wird nur zusammen mit OPCODE=*WAIT interpretiert.

Bei der Form MF=L ist nur die symbolische Adresse erlaubt, wobei symbolische Namen innerhalb einer DSECT ausgeschlossen sind, da ihre Adresse erst zur Laufzeit bekannt ist.

= adr
Ist die symbolische Adresse (Name) des Bereichs.

= (r)
Ist ein Register, das die Anfangsadresse der FPAMACC-Parameterliste enthält.

Hinweise zur Programmierung

DRV-Status

Der DRV-Status wird bei Änderungen in dem Feld FACCDS zur Verfügung gestellt. Er steht nach der ersten Ein-/Ausgabe zur Verfügung.

Auftragsendemeldung bei EVENTING

FASTPAM meldet im Zusammenhang mit dem Eventing Mechanismus (OPEN, Operand EVENTNG=*YES) das Auftragsende über das Feld FACCREQ der FPAMACC-Parameterliste. Es gibt zwei Fälle:

– FACCREQ = X'00' = FACCTERM bedeutet „Auftrag beendet“.
– FACCREQ = X'FF' = FACCACTV bedeutet „Auftrag noch nicht beendet“.

Das Feld FACCREQ wird bei Auftragsannahme mit dem Wert FACCACTV (Auftrag aktiv) und bei Ein-/Ausgabeende asynchron von einem Systemprozess mit dem Wert FACCTERM (Auftrag beendet) versorgt. Deshalb darf das Feld FACCREQ nicht mit einem schreibenden Assemblerbefehl abgefragt werden.
Zum Verlust der Endemeldung kann z.B. eine Abfrage mit folgendem Assemblerbefehl führen: OC FACCREQ,FACCREQ
Problemlos ist : CLI FACCREQ,0

Wichtig

Zwischen Auftragserteilung und Auftragsendemeldung darf auf das Feld FACCREQ nicht mit einem schreibenden Maschinenbefehl zugegriffen werden.
Alle Returninformationen (Returncode, DRV-Status) dürfen erst ausgewertet werden, nachdem die Auftragsendemeldung für sich abgefragt worden ist. Falsch wäre es z.B., den Inhalt der FPAMACC Parameterliste an eine andere Speicherstelle zu übertragen und auf dieser Kopie die entsprechenden Aktionen durchzuführen.

Der Auftrag kann sowohl synchron als auch asynchron beendet worden sein, wenn das Anwenderprogramm die Kontrolle wieder erhält. Ob das System nun ein Signal an die Börse gesendet hat, erfährt der Anwender aus dem Subcode2, der allerdings erst abgefragt werden darf, wenn der Auftrag als beendet gemeldet worden ist.

Synchrone Aufträge sind immer beendet, wenn das Anwenderprogramm die Kontrolle wieder erhält. Das Feld FACCREQ wird trotzdem sinngemäß versorgt.

Beispiel: Auftragsendebehandlung bei Eventing

```
*-------------------------------------------------------------------*
* Die Ereigniskennung, das FASTPAM-Environment und der IO-Area-Pool  *
* sollen an dieser Stelle schon eingerichtet und die Datei           *
* mit EVENT=*YES geöffnet sein.                                      *
*-------------------------------------------------------------------*

       FPAMACC MF=M,OPCODE=*READ,....
       FPAMACC MF=E,...
       CLI   FACCREQ,FACCTERM        Auftrag schon beendet?
       BE    TERM
*-------------------------------------------------------------------*
* Auftrag noch nicht beendet                                        *
* tue etwas anderes                                                 *
*-------------------------------------------------------------------*
:
          :
*-------------------------------------------------------------------*
* warte mit SOLSIG (Auftrag nicht synchron beendet!)                 *
*-------------------------------------------------------------------*
B     SOLS
TERM     CLI   FACCSR2,FACCSYTE
*-------------------------------------------------------------------*
* Auftrag synchron beendet                                          *
* kein SOSLIG!!                                                     *
*-------------------------------------------------------------------*
BE    WEITER
SOLS     SOLSIG ...
WEITER   EQU   *
          :
          :
```

Fehlerbehandlung bei Parameterlistenkettung

Stellt FASTPAM bei der Bearbeitung einer Auftragskette einen Fehler fest (sei es bei der Parameterüberprüfung oder bei der folgenden Auftragsabarbeitung), so weist FASTPAM alle folgenden Aufträge in der Kette mit dem Returncode „CHAIN_ERROR“ zurück.
Der Returncode „FACCPNAC“ (WAIT auf nicht-aktiven Ein-/Ausgabepfad) gilt dabei jedoch nicht als Fehler, da er bei synchron terminierter Ein-/Ausgabe (Caching) im normalen, korrekten Programmlauf vorkommt. Die Kette wird daher bei „FACCPNAC“ nicht abgebrochen. Daher vereinfacht sich die Fehlerabfrage.

– Bei Ketten synchroner Aufträge genügt es, den Returncode des letzten Kettenmitglieds abzufragen, um sicherzustellen, dass alle Aufträge erfolgreich ausgeführt wurden.
– Bei Ketten asynchroner Ein-/Ausgaben mit darangeketteten WAIT-Aufträgen genügt es, den Returncode des letzten WAIT-Auftrags sowie den Returncode der zugehörigen Ein-/Ausgabe abzufragen. Sind beide „SUCCESSFUL“, so ist sichergestellt, dass die vorangehenden bereits abgeschlossenen Aufträge ebenfalls erfolgreich ausgeführt wurden.
– Bei Ketten asynchroner Ein-/Ausgaben mit Eventing müssen allerdings alle Returncodes gesondert abgefragt werden, da die Auftragsbeendigungen unabhängig voneinander erfolgen.

Returncodes

Returncodes sind erst dann gültig, wenn der jeweilige Auftrag beendet ist. Returncodes werden im Header der Parameterliste (Standardheader) hinterlegt (vgl. Abschnitt „Layout der Parameterliste“ auf Seite 543):

– Der Main Returncode in einem Halbwort mit dem Namen FACCMRET.
– Der Subcode1 in einem Byte mit dem Namen FACCSR1.
  Subcode1 beschreibt Fehlerklassen, die es dem Aufrufer ermöglichen, auf Fehlerklassen zu reagieren. Der Aufrufer kann sich sowohl am Maincode als auch am Subcode1 orientieren. (Es wird empfohlen, sich am Subcode1 zu orientieren, da man damit versionsunabhängig bleibt.)
– Der Subcode2 in einem Byte mit dem Namen FACCSR2.
  Der Subcode2 spezifiert einzelne Maincodes genauer. Der Subcode2 hat beim Makro FPAMACC nur bei asynchronen Aufträgen eine Bedeutung und gibt für jeden Returncode an, ob der Auftrag bereits synchron beendet wurde. Dies gilt auch im Fehlerfall.

Die mit der C- oder D-Form des Makros generierten Feldnamen und EQU-Anweisungen für die Returncodes beginnen standardmäßig mit der Zeichenfolge FPAM, die durch PREFIX und MACID geändert werden kann.

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle „Standardheader“ auf Seite 873 entnommen werden.

Falls Returncodes nicht im Header abgelegt werden können (z.B. wenn er nicht zugreifbar ist), wird das aufrufende Programm mit einer Fehlermeldung beendet. Falls der Anwender das STXIT-Ereignis „nicht behebbarer Programmfehler“ definiert hat, wird dieser STXIT aktiviert.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

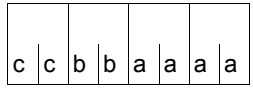
– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Im Folgenden werden die Maincodes den entsprechenden Subcode1 Klassen zugeordnet und mittels Subcode2 genauer beschrieben.

Hinweise

– Die Fehleranzeigen sind in den entsprechenden Systemmeldungen mit dem Meldungsschlüssel DFPaaaa aufgeführt (aaaa=Maincode). Meldungstexte können mit dem Kommando bzw. der Standardanweisung HELP-MSG-INFORMATION ausgegeben werden.
– Alle an FASTPAM übergebenen Adressen müssen bereinigte 31-Bit-Adressen sein. Insbesondere darf das Bit 32 nicht gesetzt sein, da dies sonst als zur Adresse gehörend betrachtet wird.

Standard-header:



Über die Ausführung des Makros FPAMACC wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAIN-CODE):

X'cc'	**X'bb'**	**X'aaaa'**	**Erläuterung**
X'00'			Synchrone Beendigung. In diesem Fall wird eine darauf folgende *WAIT Operation auf diesen Ein-/Ausgabepfad mit dem Returncode FACCPNAC (PATH NOT ACTIVE) beantwortet und bei Eventing kein Signal vom System gesendet.
X'01'			Asynchrone Beendigung. Bei EVENTNG=*NO muss der Auftrag mit *WAIT beendet werden und bei EVENTNG=*YES wird vom System ein Signal an die Auftragsendebörse gesendet.
	X'00'	X'0000'	Funktion erfolgreich ausgeführt.
	X'01'		Die Funktion konnte nicht ausgeführt werden, da der entsprechende Operand falsch spezifiziert wurde.
	X'01'	X'00C8'	Funktion nicht ausgeführt. Ungültige OPEN-ID
	X'01'	X'00C9'	Funktion nicht ausgeführt. Ungültige Adresse des Ein-/Ausgabepuffers
	X'01'	X'00CA'	Funktion nicht ausgeführt. Ungültige Blockangabe
	X'01'	X'00CB'	Funktion nicht ausgeführt. Ungültige WAITLST-Angabe
	X'01'	X'00CE'	Funktion nicht ausgeführt. Ungültige Blocknummer
	X'01'	X'00CF'	Funktion nicht ausgeführt. Ungültiger Operationscode
	X'02'		Funktion nicht ausgeführt. Aufgerufene Funktion nicht verfügbar.
	X'03'		Funktion nicht ausgeführt. Schnittstellen-Version wird nicht unterstützt.
	X'20'		Interner Fehler
	X'20'	X'0028'	Funktion nicht ausgeführt. Systemfehler. Systemdiagnose verständigen.
	X'40'		CORRECT AND RETRY

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'0037'	Systembetriebsmittelengpass Maßnahme: Systemverwaltung verständigen.
	X'40'	X'00C7'	Ungültige CFID angegeben. Maßnahme: Dateinamen im Programm korrigieren.
	X'40'	X'012C'	Ein-Ausgabefehler Maßnahme: Systemverwaltung verständigen.
	X'40'	X'012D'	Auf diesem Pfad ist noch eine Ein-/Ausgabe aktiv.
	X'40'	X'012E'	Nur bei OPCODE=*WAIT: Auf diesem Pfad ist keine Ein-/Ausgabe aktiv. Dies kann auch bei mit asynchronen Aufträgen *WAIT Operationen vorkommen, wenn die Ein-/Ausgabe synchron beendet werden konnte. Bei diesem Returncode wird die Auftragskette nicht abgebrochen.
	X'40'	X'012F'	Auftrag wurde nicht ausgeführt, da bei einem dazu geketteten Auftrag ein Fehler auftrat. Maßnahme: Fehler in der Kette suchen.
	X'40'	X'0133'	OPCODE=*WAIT bei EVENTNG=*YES nicht erlaubt.
	X'40'	X'0134'	Warten auf eine Ein-/Ausgabe eines anderen Tasks ist nicht erlaubt.
	X'40'	X'0140'	Ein-/Ausgabe hinter das Dateiende. Im Gegensatz zu UPAM erfolgt bei diesem Returncode keine einzige Ein-/Ausgabe.
	X'40'	X'0141'	Bei einer Sekundärallokierung konnte auf der Platte kein Platz mehr zugewiesen werden. Maßnahme: Systemverwaltung verständigen.
	X'40'	X'0142'	Benutzerkennung voll. Maßnahme: Dateien löschen oder Systemverwaltung verständigen.
	X'40'	X'0143'	PVS nicht attached. Maßnahme: Systemverwaltung verständigen.
	X'40'	X'0144'	Im Katalog können keine neuen Dateien aufgenommen werden. Maßnahme: Dateien löschen oder Systemverwalter verständigen.
	X'40'	X'0145'	Beim Zugriff auf eine Datei im Modus SHARUPD=YES wurde festgestellt, dass die Dateigröße den Wert von 32 GB übersteigt, beim OPEN für diese Datei wurde aber ein Überschreiten von 32 GB nicht erlaubt.
	X'40'	X'014A'	Wegen fehlender Klasse-4-Extent-Liste kann keine Sekundärallokierung vorgenommen werden. Das Fehlen der Extent-Liste weist darauf hin, dass die Systemverwaltung das Kommando REPAIR-DISK-FILES bzw. REMOVE-FILE-ALLOCATION auf die gerade geöffnete Datei abgesetzt hat. Maßnahme: Systemverwaltung verständigen.

Beispiele

Beispiel 1: Fehler während der Parameterüberprüfung

Bevor der erste Auftrag in einer Kette bearbeitet wird, überprüft FASTPAM die Parameter aller Kettenmitglieder. Dabei wird ein Parameterfehler „INVALID_<parameter>“ festgestellt.

(In den folgenden Diagrammen sollen die Kästchen FPAMACC-Parameterlisten darstellen. Darunter stehen die jeweiligen Returncodes.)

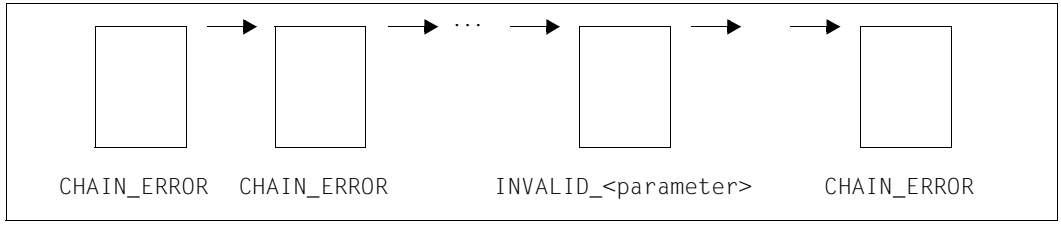


Bild 8: FPAMACC-Makro: Fehler während der Parameterüberprüfung

Beispiel 2: Fehler nach der Parameterüberprüfung

– Asynchrone Ein-/Ausgaben mit Eventing:
  Tritt in einer Kette asynchroner Ein-/Ausgaben mit Eventing ein Fehler auf, nachdem alle Ein-/Ausgaben angestartet worden sind (z.B. ein Ein-/Ausgabefehler „IO_ERROR“), wird jeder Auftrag gesondert behandelt. „CHAIN_ERROR“ kann nicht auftreten.

  Jede Ein-/Ausgabe geht auf eine Datei, die mit „Eventing“ geöffnet ist. READ/WRITE kann beliebig vertauscht werden:

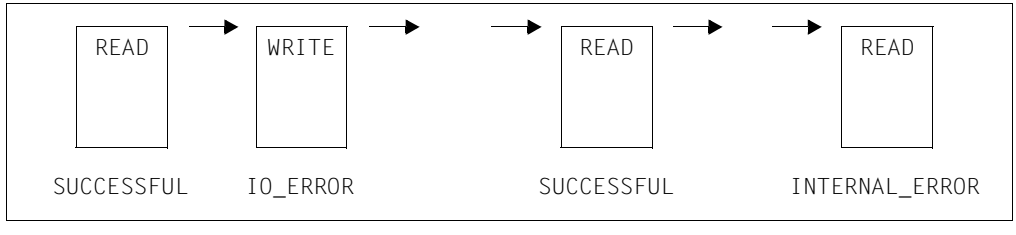


Bild 9: FPAMACC-Makro: Fehler nach der Parameterüberprüfung (asynchrone Ein-/Ausgabe mit Eventing)

– Asynchrone Ein-/Ausgaben mit „WAIT“:
In einer Kette asynchroner Ein-/Ausgaben mit dahinter geketteten WAIT-Operationen tritt bei der n-ten Ein-/Ausgabe ein Fehler auf („IO_ERROR“). Die zugehörige WAIT-Operation, bei der der Fehler bemerkt wird, wird trotzdem als erfolgreich betrachtet. Da die Verarbeitung nach einem Fehler abgebrochen wird, fehlen jedoch die darauffolgenen WAIT-Operationen auf die Ausgaben 1 bis (n-1).

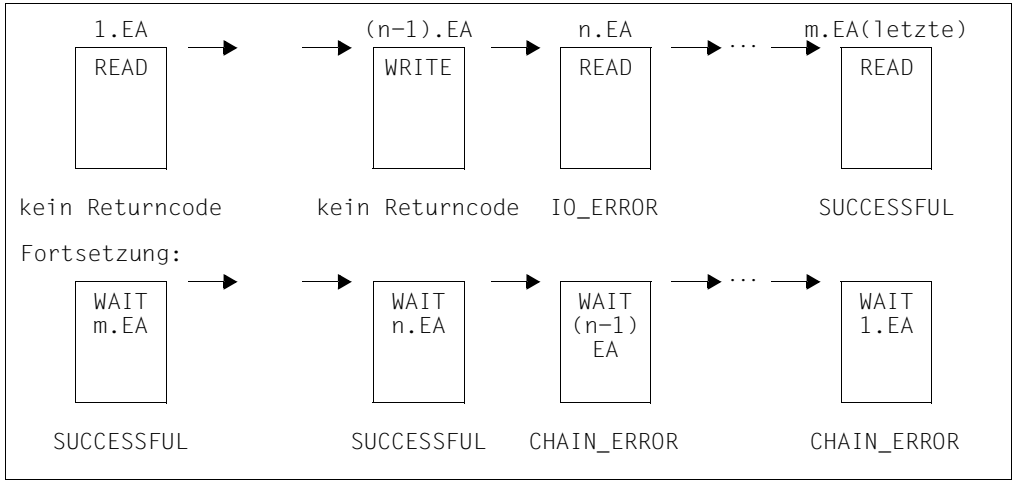


Bild 10: FPAMACC-Makro: Fehler nach der Parameterüberprüfung (asynchrone Ein-/Ausgabe mit WAIT)

Layout der Parameterliste

Von einem FPAMACC Makroauruf wird folgende Parameterliste abgesetzt:

```
FPAMACC MF=D
1          STACK  PRINT
1          PRINT  NOGEN
2               *,##### PREFIX=F, MACID=ACC #####
1          #INTF REFTYPE=REQUEST,INTNAME=FPAMACC,INTCOMP=001
1 FACCPA   DS    OF      BEGIN of PARAMETERAREA           _INOUT
1          FHDR  MF=(C,FACC),EQUATES=YES
2          DS    0A
2 FACCFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
2 *
2 FACCIFID DS    0A              0   INTERFACE IDENTIFIER
2 FACCFCTU DS    AL2             0   FUNCTION UNIT NUMBER
2 *                                  BIT 15    HEADER FLAG BIT,
2 *                                  MUST BE RESET UNTIL FURTHER NOTICE
2 *                                  BIT 14-12 UNUSED, MUST BE RESET
2 *                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
2 FACCFCT  DS    AL1             2   FUNCTION NUMBER
2 FACCFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
2 *
2 FACCRET  DS    0A              4   GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 FACCSRET DS    0AL2            4   SUB RETURN CODE
2 FACCSR2  DS    AL1             4   SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 FACCR2OK EQU   X'00'                All correct, no additional info
2 FACCR2NA EQU   X'01'                Successful, no action was necessary
2 FACCR2WA EQU   X'02'                Warning, particular situation
2 FACCSR1  DS    AL1             5   SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A    X'00'           FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B    X'01' - X'1F'   PARAMETER SYNTAX ERROR
2 * CLASS C    X'20'           INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D    X'40' - X'7F'   NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E    X'80' - X'82'   WAIT AND RETRY
2 *
2 FACCRFSP EQU   X'00'                FUNCTION SUCCESSFULLY PROCESSED
2 FACCRPER EQU   X'01'                PARAMETER SYNTAX ERROR
2 *  3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
```

```
2 FACCRFNS EQU   X'01'              CALLED FUNCTION NOT SUPPORTED
2 FACCRFNA EQU   X'02'              CALLED FUNCTION NOT AVAILABLE
2 FACCRVNA EQU   X'03'              INTERFACE VERSION NOT SUPPORTED
2 *
2 FACCRAER EQU   X'04'              ALIGNMENT ERROR
2 FACCRIER EQU   X'20'              INTERNAL ERROR
2 FACCRCAR EQU   X'40'              CORRECT AND RETRY
2 *  2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 FACCRECR EQU   X'41'              SUBSYSTEM (SS) MUST BE CREATED
2 *                                 EXPLICITELY BY CREATE-SS
2 FACCRECN EQU   X'42'              SS MUST BE EXPLICITELY CONNECTED
2 *
2 FACCRWAR EQU   X'80'              WAIT FOR A SHORT TIME AND RETRY
2 FACCRWLR EQU   X'81'                   "     LONG       "
2 FACCRWUR EQU   X'82'              WAIT TIME IS UNCALCULABLY LONG
2 *                                 BUT RETRY IS POSSIBLE
2 *  2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 FACCRTNA EQU   X'81'              SS TEMPORARILY NOT AVAILABLE
2 FACCRDH  EQU   X'82'              SS IN DELETE / HOLD
2 *
2 FACCMRET DS    0AL2           6   MAIN RETURN CODE
2 FACCMR2  DS    AL1            6   MAIN RETURN CODE 2
2 FACCMR1  DS    AL1            7   MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XXYYYY')
2 *
2 FACCRLNK EQU   X'FFFF'             LINKAGE ERROR / REQ. NOT PROCESSED
2 FACCFHL  EQU   8              8   GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *
1 * MAINCODE
1 *
1 FACCMFSP EQU  X'0000'       SUCCESSFUL_PROCESSING              = 0
1 FACCMIER EQU  X'0028'       INTERNAL_ERROR                     = 40
1 FACCSRES EQU  X'0037'       SHORTAGE_OF_RESOURCES              = 55
1 FACCICFI EQU  X'00C7'       INVALID_CFID                       = 199
1 FACCIOPI EQU  X'00C8'       INVALID_OPEN_ID                    = 200
1 FACCIIOA EQU  X'00C9'       INVALID ADDRESS OF IOAREA          = 201
1 FACCIBLK EQU  X'00CA'       INVALID_BLOCK                      = 202
1 FACCILAW EQU  X'00CB'       INVALID LIST ADDRESS FOR WAIT      = 203
1 FACCIBL# EQU  X'00CE'       INVALID_BLOCK_#                    = 206
1 FACCIOP  EQU  X'00CF'       INVALID_OPCODE                     = 207
1 FACCIOER EQU  X'012C'       IO_ERROR = 300
1 FACCPACT EQU  X'012D'       PATH_ACTIVE                        = 301
1 FACCPNAC EQU  X'012E'       PATH_NOT_ACTIVE (WAIT ONLY)        = 302
1 FACCCHE  EQU  X'012F'       CHAIN_ERROR                        = 303
1 FACCWTEV EQU  X'0133'       WAIT_AND_EVENTING                  = 307
1 FACCWTNS EQU  X'0134'       WAIT_NOT_BY_SAME_TASK              = 308
```

```
1 FACCEOF  EQU  X'0140'       END_OF_FILE                        = 320
1 FACCNDSA EQU  X'0141'       NO_DISC_SPACE_AVAILABLE            = 321
1 FACCUIDE EQU  X'0142'       USER_ID_EXHAUSTED                  = 322
1 FACCPVNA EQU  X'0143'       PUBLIC_VOLUME_NOT_ATTACHED         = 323
1 FACCCEFL EQU  X'0144'       CATALOG_ENTRY_FULL                 = 324
1 FACCLFNS EQU  X'0145'       LARGE_FILE_NOT_SPECIFIED           = 325
1 FACCSAVY EQU  X'014A'       SYSTEM_ADMINISTRATOR_VERIFY        = 330
1 *
1 * SUB RETURN CODE2
1 *
1 FACCSYTE EQU  X'00'         SYNCHRONEOUS TERMINATION
1 FACCASTE EQU  X'01'         ASYNCHRONEOUS TERMINATION
1 *
1 * FPAMACC FUNCTIONS:
1 *
1 FACCACCF EQU  7             ACCESS FILE
1 *
1 * OUTPUT PARAMETER
1 *
1          DS   XL2           RESERVED
1 FACCDS   DS   X             DRV STATUS
1 FACCREQ  DS   X             REQUEST STATUS
1 FACCTERM EQU  X'00'         REQUEST TERMINATED
1 FACCACTV EQU  X'FF'         REQUEST ACTIVE
1 *
1 * INPUT PARAMETER
1 *
1 FACCOPID DS   F             OPEN-ID
1 FACCIOA  DS   A             ADDRESS OF IOAREA
1 FACCBLK  DS   F             BLOCK WITHIN FILE
1 FACCLAW  DS   A             LIST ADDRESS FOR WAIT OPERATION
1 FACCCHLA DS   A             ADDRESS OF CHAINED LIST
1 FACCPOCO DS   FL2           POSTCODE
1 FACCBLK# DS   FL1           BLOCK NUMBER
1 FACCOP   DS   AL1           OPCODE
1 FACCREAD EQU  1             READ
1 FACCWRIT EQU  2             WRITE
1 FACCRDWT EQU  3             READ AND WAIT
1 FACCWRWT EQU  4             WRITE AND WAIT
1 FACCRDEQ EQU  5             READ AND EQUALIZE
1 FACCWAIT EQU  6             WAIT
1          DS   0F
1 FACC#    EQU   *-FACCPA LENGTH of PARAMETERAREA
```

Beispiel

Erstellen einer Datei mit FASTPAM

```
FPAMTEST START
         BALR  10,0
         USING *,10
         USING FPAMD,9
         LA    9,FPAMPL               R9 -> FPAMSRV-Parameterliste
*-------------------------------------------------------------------*
* Speicher für die ACCESS-Parameterlisten                           *
*-------------------------------------------------------------------*
REQM
         LTR   15,15
         BNZ   ERROR
         LR    8,1                   R8 -> FPAMACC-Parameterliste
*-------------------------------------------------------------------*
* Speicher für den IOAREA-POOL                                      *
*-------------------------------------------------------------------*
REQM  30
         LTR   15,15
         BNZ   ERROR
         LR    7,1                   R7 -> IOAREA-POOL
*-------------------------------------------------------------------*
* ENABLE ENVIRONMENT                                                *
*-------------------------------------------------------------------*
FPAMSRV MF=M,PARAM=FPAMPL,ACCLSTS=(8)
         FPAMSRV MF=E,PARAM=FPAMPL
         CLI   FPAMSR1,FPAMRFSP
         BE    ENAIPO
         CLI   FPAMSR1,FPAMRCAR
         BNE   ERROR
         CLC   FPAMMRET,=Y(FPAMNORE)
         BNE   ERROR
*-------------------------------------------------------------------*
* Behandlung des Fehlers 'RESIDENT SPACE NOT AVAILABLE'             *
* evt. nur Meldung ausgeben und weitermachen                        *
*-------------------------------------------------------------------*
*              .
*              .
*              .
*-------------------------------------------------------------------*
* ENABLE IOAREA-POOL                                                *
*-------------------------------------------------------------------*
ENAIPO   FPAMSRV MF=M,PARAM=FPAMPL,FCT=*ENAIPO,IPONAME='IOAREA',      -
               IPOADDR=((7),0),IPOSIZE=30
         FPAMSRV MF=E,PARAM=FPAMPL
         CLI   FPAMSR1,FPAMRFSP
```

```
         BE    OPEN
         CLI   FPAMSR1,FPAMRCAR
         BNE   ERROR
         CLC   FPAMMRET,=Y(FPAMNORE)
         BNE   ERROR
*-------------------------------------------------------------------*
* Behandlung des Fehlers 'RESIDENT SPACE NOT AVAILABLE'             *
* evt. nur Meldung ausgeben und weitermachen
*-------------------------------------------------------------------*
*              .
*              .
*              .
*-------------------------------------------------------------------*
* Eröffnen der Datei mit OUTIN                                      *
*-------------------------------------------------------------------*
OPEN     FPAMSRV MF=M,PARAM=FPAMPL,FCT=*OPEN,FILE='TESTFILE',       -
               MODE=*OUTIN,SHARUPD=*YES,BLKSIZE=1
         FPAMSRV MF=E,PARAM=FPAMPL
         CLI   FPAMSR1,FPAMRFSP
         BNE   ERROR
*-------------------------------------------------------------------*
* Schreibe nummerierte Blöcke in die Datei                          *
*-------------------------------------------------------------------*
LA    6,1
         LR    4,8                 R4 -> 1. Parameterliste
         USING ACCESSD,4
         LR    3,8                 R3 -> 1. Parameterliste
         LA    2,30                 Schleifenzähler
CYCL1    DS    OF
         ST    6,0(7)
         MVC   0(FACC#,4),FACCPL
         C     2,=A(1)
         BNE   NOTLAST
* IN DER LETZTEN FPAMACC-PARAMETERLISTE KEINE KETTUNG MEHR
         L     3,FFFFFFFF
         B     NEXT
NOTLAST  EQU   *
         A     3,=A(FACC#)          R3 -> nächste Parameterliste
NEXT     EQU   *
         FPAMACC MF=M,PARAM=(4),OPENID=FPAMOPID,BLOCK=(6),          -
               IOAREA=(7),CHAIN=(3)
         A     6,=A(1)
         LR    4,3                 R4 -> nächste Parameterliste
         BCT   2,CYCL1
         FPAMSRV MF=E,PARAM=(8)
* Fehlerauswertung der Parameterlisten                              *
         LR    3,8                 R3 -> 1. Parameterliste
         USING ACCESSD,3
```

```
         LA    2,30                  Schleifenzähler
CYCL2    DS    OF
         CLI   FACCSR1,FACCRFSP
         BNE   ERROR
         A     3,=A(FACC#)           R3 -> nächste Parameterliste
         BCT   2,CYCL2
*
*-------------------------------------------------------------------*
* Schließen der Datei                                               *
*-------------------------------------------------------------------*
FPAMSRV MF=M,PARAM=FPAMPL,FCT=*CLOSE
         FPAMSRV MF=E,PARAM=FPAMPL
         CLI   FPAMSR1,FPAMRFSP
         BNE   ERROR
*-------------------------------------------------------------------*
* DISABLE IOAREA-POOL                                               *
*-------------------------------------------------------------------*
FPAMSRV MF=M,PARAM=FPAMPL,FCT=*DISIPO
         FPAMSRV MF=E,PARAM=FPAMPL
         CLI   FPAMSR1,FPAMRFSP
         BNE   ERROR
*-------------------------------------------------------------------*
* DISABLE ENVIRONMENT                                               *
*-------------------------------------------------------------------*
FPAMSRV MF=M,PARAM=FPAMPL,FCT=*DISENV
         FPAMSRV MF=E,PARAM=FPAMPL
         CLI   FPAMSR1,FPAMRFSP
         BNE   ERROR
*-------------------------------------------------------------------*
* Speicherfreigabe für den IOAREA-POOL                              *
*-------------------------------------------------------------------*
RELM  30,(7)
*-------------------------------------------------------------------*
* Speicherfreigabe für die ACCESS-Parameterlisten                   *
*-------------------------------------------------------------------*
RELM  1,(8)
*
ERROR    DS   OY
         TERM
*
FPAMPL   FPAMSRV  MF=L,FCT=*ENAENV,ENVNAME='TESTENV',ACCNUMB=30,      -
               MAXIOLN=*MINI,EVENTNG=*NO
FPAMD    FPAMSRV  MF=D
FACCPL   FPAMACC MF=L,LEN=1,                                          -
               OPCODE=*WRITE_WAIT
FFFFFFFF DC    X'FFFFFFFF'
*
         END
```

FPAMSRV – FASTPAM-Verwaltungsaufrufe formulieren

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form, siehe Seite 870)

Allgemeines

Zunächst werden in einer Formatübersicht sämtliche Funktionsoperanden des Makros FPAMSRV dargestellt. Unabhängig von der angegebenen Funktion (Operand FCT) können bei einem FPAMSRV-Makro **alle** Operanden spezifiziert werden. Welche Operanden ausgewertet werden, hängt von der aktuellen FPAMSRV-Funktion ab. Im Anschluss an die Formatübersicht werden kurz die Funktionen des FPAMSRV-Makros aufgelistet. Das Format und die Operanden, die bei den einzelnen Funktionen ausgewertet werden, werden pro Funktionseinheit beschrieben.
Operandenwerte, die nicht Adress- und nicht Register-Angaben sind, werden in der Operandenbeschreibung als „direkte Angabe“ bezeichnet.
Die „direkte Angabe“ ist in der Operandenbeschreibung immer auch dann aufgeführt, wenn sie nur formal möglich ist, weil der Anwender den Wert bei dieser Angabe zum Programmierzeitpunkt noch nicht kennen kann (z.B. den Wert einer vom System vergebenen Kurzkennung).

Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

Parameterliste

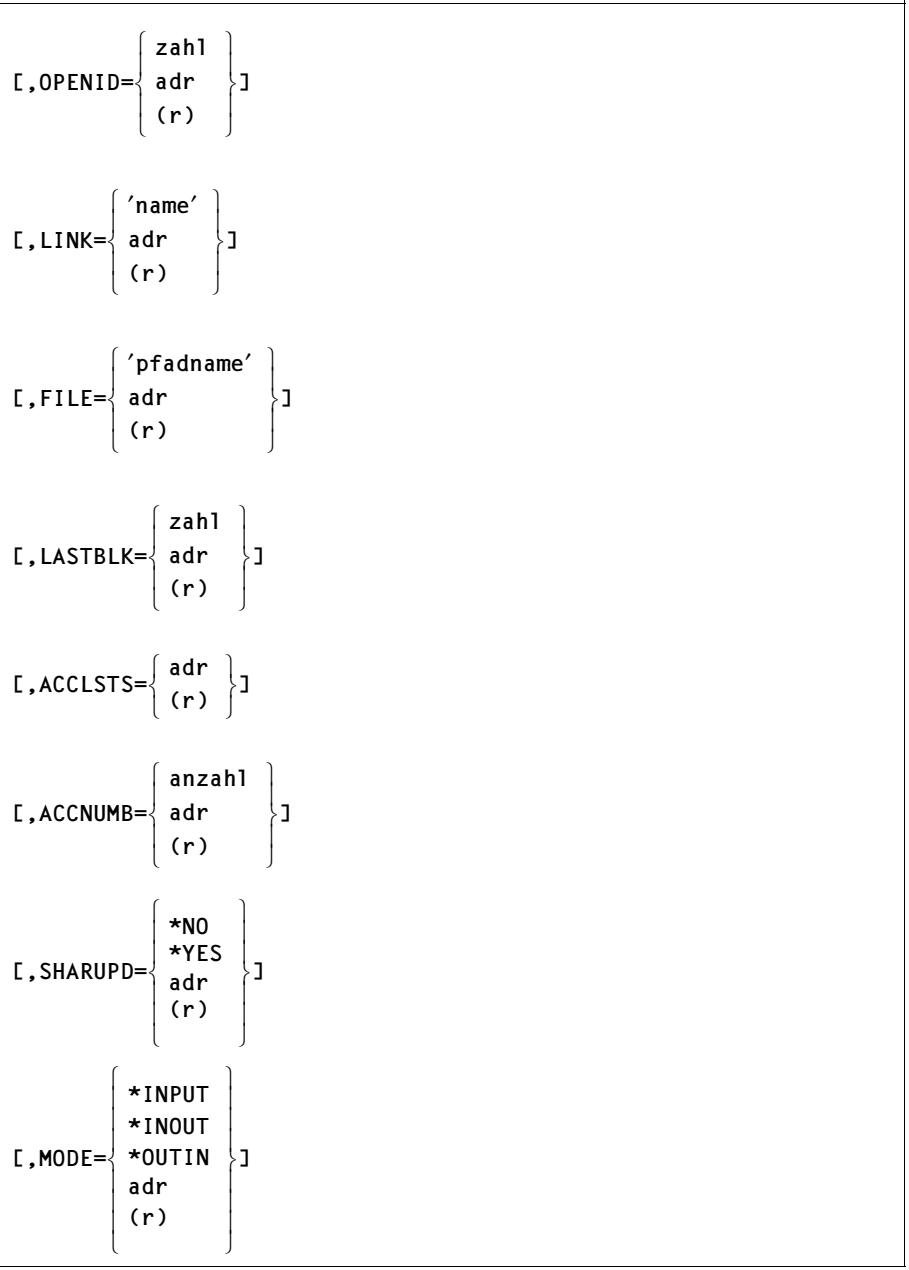
Die Parameterliste des Makros enthält einen Header, dessen Felder beim Aufbau der Parameterliste mithilfe der L-Form automatisch versorgt werden.

Wird eine Parameterliste mit der D-Form oder C-Form dynamisch aufgebaut, ist sie zuvor durch eine mithilfe der L-Form erzeugten Parameterliste zu initialisieren. Nur auf diese Weise ist eine korrekte Versorgung des Header einer Parameterliste gewährleistet.

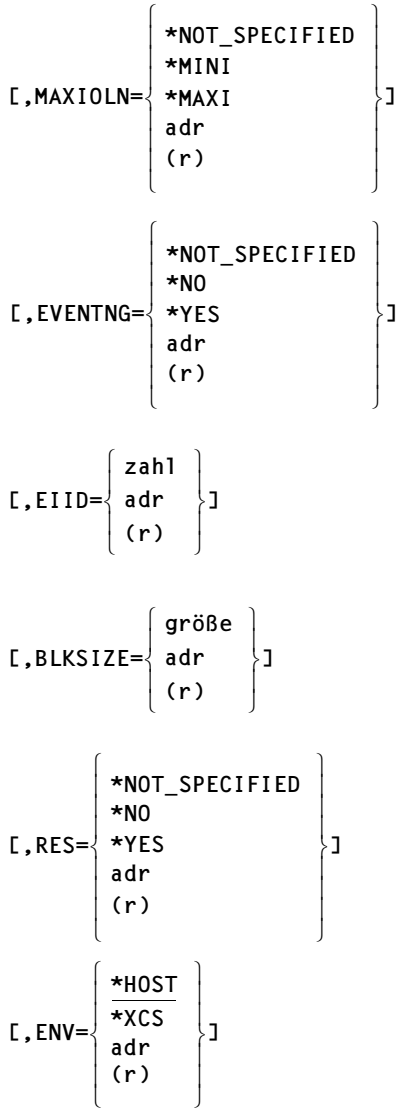
Format

Operation	Operanden
FPAMSRV	[,FCT={*ENAENV | *ENAIPO | *OPEN | *CLOSE | *DISIPO | *DISENV | adr | (r)}] [,ENVNAME={'name' | adr | (r)}] [,IPONAME={'name' | adr | (r)}] [,IPOADDR=(elem1,elem2)] [,IPOSIZE={größe | adr | (r)}] [,ENVID={zahl | adr | (r)}] [,IPOID={zahl | adr | (r)}]

(Teil 1 von 4)

Operation	Operanden
	

(Teil 2 von 4)

Operation	Operanden
	

(Teil 3 von 4)

Operation	Operanden
	[,LARGE_FILE={ *FORBIDDEN / *ALLOWED / adr / (r) }]
	MF=L
	MF=E,PARAM={ adr / (r) }
	MF=D[,PREFIX={ F / pre }]
	MF={ C / M }[,PREFIX={ F / pre }][,MACID={ PAM / macid }]

(Teil 4 von 4)

Funktionen

Funktion	Kurzbeschreibung	siehe
FCT = *ENAENV	FASTPAM-Environment einrichten oder den Aufrufer an ein bestehendes anschließen	Seite 554
FCT = *ENAIPO	FASTPAM-IO-Area-Pool einrichten oder den Aufrufer an einen bestehenden anschließen	Seite 563
FCT = *OPEN	PAM-Datei eröffnen	Seite 570
FCT = *CLOSE	PAM-Datei schließen	Seite 579
FCT = *DISIPO	Verbindung zum FASTPAM-IO-Area-Pool lösen und ggf. FASTPAM-IO-Area-Pool abbauen	Seite 582
FCT = *DISENV	Verbindung zum FASTPAM-Environment lösen u. ggf. FASTPAM-Environment abbauen	Seite 585

Hinweis

Alle an FASTPAM übergebenen Adressen müssen bereinigte 31-Bit-Adressen sein. Insbesondere darf das Bit 32 nicht gesetzt sein, da dies sonst als zur Adresse gehörend betrachtet wird.

Funktion ENABLE ENVIRONMENT

Es wird ein FASTPAM-Environment eingerichtet oder der Aufrufer an ein bestehendes FASTPAM-Environment angeschlossen.
In die Parameterliste wird eine Kennung eingetragen (FPAMENID), die bei den folgenden OPEN Aufrufen zu benutzen ist. Wenn dabei dieselbe Parameterliste benutzt wird, ist die Kennung bereits eingetragen und muss dann nicht eigens versorgt werden.

Von der Funktion ENAENV werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet.

Format FCT=*ENAENV

Operation	Operanden
FPAMSRV	[,FCT={*ENAENV | adr | (r)}] [,ENVNAME={'name' | adr | (r)}] [,ACCLSTS={adr | (r)}] [,ACCNUMB={anzahl | adr | (r)}] [,MAXIOLN={*NOT_SPECIFIED | *MINI | *MAXI | adr | (r)}]

(Teil 1 von 2)

Operation	Operanden
	[,EVENTNG={ *NOT_SPECIFIED / *NO / *YES / adr / (r) }]
	[,EIID={ zahl / adr / (r) }]
	[,RES={ *NOT_SPECIFIED / *NO / *YES / adr / (r) }]
	MF=L
	MF=E,PARAM={ adr / (r) }
	MF=D[,PREFIX={ F / pre }]
	MF={ C / M }[,PREFIX={ F / pre }][,MACID={ PAM / macid }]

(Teil 2 von 2)

Operandenbeschreibung

ACCLSTS
Gibt die auf 4 KByte ausgerichtete Anfangsadresse des Bereichs an, der sämtliche, hintereinander liegenden FPAMACC-Parameterlisten enthält. Ist dieser Bereich resident angelegt, darf er sich nicht mit dem Parameterlistenbereich anderer Environments, mit einem IO-Area-Pool oder einem DIV-Fenster überlappen.

Liegt der Bereich in einem Memory-Pool, muss dieser mit dem Makro ENAMP, Operand FIXED=YES angelegt worden sein.

Bei der Form MF=L ist nur die symbolische Adresse erlaubt, wobei symbolische Namen innerhalb einer DSECT ausgeschlossen sind, da ihre Adresse erst zur Laufzeit bekannt ist.

= adr
Ist die symbolische Adresse (Name) des Bereichs.

= (r)
Ist ein Register, das die Anfangsadresse des Bereichs enthält.

ACCNUMB
ACCNUMB gibt die Anzahl der Parameterlisten an, die in dem mit ACCLSTS angegebenen Bereich liegen. Dieser Bereich muss also mindestens mit der Größe ACCNUMB * (Länge der FPAMACC-Parameterliste) angefordert werden.

Läuft die Anwendung nicht mit FASTPAM-Berechtigung, ist die Anzahl der Parameterlisten auf 500 beschränkt; ansonsten ist der Höchstwert 5000.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= anzahl
Gibt die Anzahl der Parameterlisten an, die in dem mit ACCLSTS angegebenen Bereich liegen sollen ($1 \leq$ anzahl ≤ 5000).

= adr
Ist die symbolische Adresse eines 4 Byte langen Feldes, das die Anzahl der Parameterlisten als numerischen Wert enthält (binär).

= (r)
Ist ein Register, das die Anzahl der Parameterlisten als numerischen Wert enthält.

EIID
Gibt die Kurzkennung für die Ereigniskennung an, über die bei den Dateizugriffen das Auftragsende angezeigt wird. Diese Kurzkennung erhält der Anwender über den Operanden EIIDRET des Makros ENAEI (siehe Handbuch „Makroaufrufe an den Ablaufteil“ [2]). Liegen die FPAMACC-Parameterlisten in einem Memory Pool, ist darauf zu achten, dass sein Geltungsbereich nicht größer ist als der Geltungsbereich der Ereigniskennung (Returncode FPAMEISS).

Bei EVENTNG=*NO wird der Operand EIID nicht ausgewertet.

Bei der Form MF=L ist nur die direkte Angabe möglich.

= zahl
Ist der dezimale numerische Wert der Kurzbezeichnung für die Ereigniskennung.

= adr
Ist die symbolische Adresse (Name) des 4 Byte langen Feldes, das die Kurzkennung für die Ereigniskennung enthält.

= (r)
Ist ein Register, das die Kurzkennung für die Ereigniskennung enthält.

ENVNAME
Bezeichnet den Namen des Environments.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= 'name'
Ist der Name des Environments.
Namenslänge: 1 ≤ 'name' ≤ 54 Zeichen.
Namensbildung:
1. Zeichen: Buchstabe oder Sonderzeichen #, @ (für TPR-Tasks auch $).
2. – 54. Zeichen: beliebige Kombination aus der Zeichenmenge (A,...,Z,0,...,9,$,#,@).

Das erste Blank (X'40') beendet den Namen.
Der Name muss in Hochkommata eingeschlossen werden.

= adr
Ist die symbolische Adresse eines 54 Byte langen Feldes, das den Namen des Environments enthält.

= (r)
Ist ein Register, das die Adresse des Feldes mit dem Namen des Environments enthält.

EVENTNG
Legt fest, ob der Anwender bei den asynchron ausgeführten Dateizugriffen das Auftragsende über Eventing mitgeteilt bekommt (vgl. dazu Abschnitt „FASTPAM-Funktionen, Eventing Verarbeitung“, Handbuch „Einführung in das DVS“ [1]).

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *YES
Der Anwender will mit Eventing arbeiten. In diesem Fall muss die Kurzbezeichnung der Ereigniskennung (Eventitem; Operand EIID) angegeben werden. Bei den nachfolgenden OPEN-Aufrufen kann sowohl mit EVENTNG=*YES, als auch mit EVENTNG=*NO gearbeitet werden.

= *NO
In diesem Fall kann mit diesem Environment keine Datei mehr mit dem Parameter EVENTNG=*YES eröffnet werden.

Bei EVENTNG=*NO wird der Operand EIID nicht ausgewertet.

= *NOT_SPECIFIED
Das Environment besteht schon und der Teilnehmer will sich unabhängig von der entsprechenden Eigenschaft daran anschließen.

= adr
Ist die symb. Adresse eines 1 Byte langen Feldes, das den Wert für EVENTNG enthält.

= (r)
Ist ein Register, das den Wert für EVENTNG enthält.

FCT
Bestimmt die auszuführende FASTPAM-Funktion.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *ENAENV
Ist die direkte Angabe der Funktion ENABLE ENVIRONMENT.

Es wird ein FASTPAM-Environment eingerichtet, oder der Aufrufer wird an ein bereits bestehendes FASTPAM-Environment angeschlossen.
In die Parameterliste wird eine Kennung eingetragen (FPAMENID), die bei den folgenden OPEN-Aufrufen zu benutzen ist. Wenn dabei dieselbe Parameterliste benutzt wird, ist die Kennung bereits eingetragen und braucht daher nicht beachtet zu werden.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die Funktion ENABLE ENVIRONMENT enthält.

= (r)
Ist ein Register, das den Wert für die für die Funktion ENABLE ENVIRONMENT enthält.

MACID
Legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

= PAM
Voreinstellung: MACID=PAM

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MAXIOLN
Bestimmt die mit diesem Environment mögliche maximale Ein-/Ausgabelänge. Sie darf bei den Dateizugriffen nicht mehr über-, sondern nur noch unterschritten werden.
Beim Arbeiten mit residenten FASTPAM-Environments wird für die vorformatierten Ein-/Ausgabepfade zusätzlich residenter Systemspeicher (Klasse-3-Speicher) belegt, und zwar pro Ein-/Ausgabepfad
– 1 KByte bei MAXIOLN = *MINI
– 2 KByte bei MAXIOLN = *MAXI

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *MINI
Für jede FPAMACC-Parameterliste wird ein Ein-/Ausgabepfad zur Übertragung von 4 KByte angelegt.

= *MAXI
Für jede FPAMACC-Parameterliste wird ein Ein-/Ausgabepfad zur Übertragung von 32 KByte angelegt.

= *NOT_SPECIFIED
Das Environment besteht schon, und der Teilnehmer will sich unabhängig von der entsprechenden Eigenschaft daran anschließen.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das die maximale Ein-/Ausgabelänge für dieses Environment enthält.

= (r)
Ist ein Register, das den Wert für MAXIOLN enthält.

MF
Die Formen des MF-Operanden sind detailliert im Anhang, Seite 870 beschrieben.

PARAM
Bezeichnet die Adresse der Operandenliste. Der Operand wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

PREFIX
legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

= F
Voreinstellung: PREFIX=F

= pre
„pre“ ist ein ein Zeichen langes Präfix, mit dem die generierten Feldnamen und Equates beginnen sollen.

RES
Gibt an, ob das Environment resident oder nicht-resident angelegt werden soll.

= *YES
Das Environment soll resident angelegt werden. In diesem Fall wird überprüft, ob die Benutzerkennung über die FASTPAM-Berechtigung verfügt und ob die beim Programmaufruf angeforderte Anzahl residenter Seiten (Benutzerkatalog: „RESIDENT-PAGES“ bzw. „CLASSII“) für den FPAMACC-Parameterlistenbereich ausreicht. Die Größe des FPAMACC-Parameterlistenbereichs wird durch den Operanden ACCNUMB bestimmt.
Bei negativem Ergebnis erhält der Anwender die Fehlermeldung „FPAMNORE“. Das FASTPAM-Environment wird in diesem Fall nicht-resident angelegt.

= *NO
Das Environment wird nicht-resident angelegt.

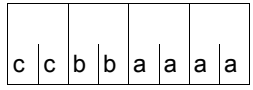
= *NOT_SPECIFIED
Das Environment besteht schon, und der Teilnehmer will sich daran anschließen, unabhängig davon, ob es resident oder nicht-resident angelegt ist.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für RES enthält.

= (r)
Ist ein Register, das den Wert für RES enthält.

*Mögliche Returncodes bei FCT=*ENAENV*

Standard-header:



Über die Ausführung des Makros FPAMSRV wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAIN-CODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Funktion erfolgreich ausgeführt.
	X'01'	X'0001'	Funktion nicht ausgeführt. Ungültiger Name des Environments
	X'01'	X'0005'	Funktion nicht ausgeführt. Ungültige Adresse der Operandenliste
	X'01'	X'0006'	Funktion nicht ausgeführt. Ungültige Anzahl von Operandenlisten
	X'01'	X'000A'	Funktion nicht ausgeführt. Die maximal erlaubte Länge des Ein-/Ausgabebereichs wurde überschritten (max. 4 KByte bzw. 32 KByte).
	X'01'	X'000B'	Funktion nicht ausgeführt. Ungültige Ereigniskurzkennung
	X'01'	X'000D'	Funktion nicht ausgeführt. Ungültige Kurzkennung des Eventings
	X'01'	X'0012'	Funktion nicht ausgeführt. – Ungültige Angabe beim Operanden RES – Angabe NOT_SPECIFIED beim Operanden RES und das spezifizierte Environment existiert noch nicht
	X'20'	X'0028'	Funktion nicht ausgeführt. Systemfehler. Systemdiagnose einschalten.
	X'40'	X'0032'	Das Environment bzw. der IO-Area-Pool konnte nicht resident angelegt werden. Zu Testzwecken kann das Anwendersystem weiterarbeiten, allerdings auf Kosten der FASTPAM-Performancevorteile. Der subcode2 spezifiziert die Ursache.
X'01'	X'40'	X'0032'	Die USERID der Task, die das Environment bzw. den IO-Area-Pool eingerichtet hat, verfügt nicht über die FASTPAM-Berechtigung. Maßnahme: Systemverwalter verständigen.
X'02'	X'40'	X'0032'	Zu kleiner Realspeicher.
X'03'	X'40'	X'0032'	Das beim Programmstart angegebene Kontingent an residentem Hauptspeicher ist überschritten.
X'04'	X'40'	X'0032'	Konnektierung an ein nicht-residentes Environment bzw. an einen nicht-residenten IO-Area-Pool.

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'05'	X'40'	X'0032'	In vorliegender FASTPAM-Version werden Datenräume (Data-Spaces) nur nicht-resident unterstützt.
	X'40'	X'0035'	Während der Ausführung der Funktionen *ENAENV/*ENAIPO läuft eine Ein-/Ausgabe auf die zu fixierenden Speicherseiten. Dieser Returncode kommt nur beim Einrichten des Environments bzw. IO-Area-Pools. Während der Konnektierung dürfen Ein-/Ausgaben laufen.
	X'40'	X'0037'	Systembetriebsmittelengpass. Maßnahme: Systemverwalter verständigen.
	X'40'	X'0038'	An das genannte FASTPAM-Betriebsmittel können sich nur TPR-Tasks konnektieren.
	X'40'	X'0039'	An das genannte FASTPAM-Betriebsmittel können sich nur Tasks einer Kennung mit FASTPAM-Privileg konnektieren.
	X'40'	X'003B'	Der Teilnehmer will sich an ein bestehendes Environment mit einem anderen Operandenwert für ACCLSTS konnektieren.
	X'40'	X'003C'	Der Teilnehmer will sich an ein bestehendes Environment mit einem anderen Operandenwert für ACCNUMB konnektieren.
	X'40'	X'003F'	Der Teilnehmer will sich an ein bestehendes Environment mit einem anderen Operandenwert für MAXIOLN konnektieren.
	X'40'	X'0040'	Der Teilnehmer will sich an ein bestehendes Environment mit einem anderen Operandenwert für EVENTNG konnektieren.
	X'40'	X'0041'	Der Teilnehmer will sich an ein bestehendes Environment mit einer anderen Ereigniskennung konnektieren.
	X'40'	X'0046'	Der Anwender will sich an ein bestehendes Environment anschließen, ist aber nicht an die zugehörige Ereigniskurzkennung angeschlossen.
	X'40'	X'0047'	Der Gültigkeitsbereich der Ereigniskennung ist kleiner als der Gültigkeitsbereich des FPAMACC-Parameterlistenspeichers.
	X'40'	X'0048'	Die Task ist an das Environment bereits konnektiert.
	X'40'	X'004A'	Der angegebene Anwenderspeicher überlappt sich mit einem DIV-Fenster. Dieser Returncode kommt nur im residenten Fall.
	X'40'	X'004B'	Der angegebene Anwenderspeicher überlappt sich mit einem bereits verwendeten FASTPAM-Anwenderspeicher. Dieser Returncode wird nur ausgegeben, wenn mit residentem Environment bzw. mit residentem IO-Area-Pool gearbeitet wird.
	X'40'	X'004C'	Der Speicher für die FPAMACC-Parameterlisten ist nicht vollständig angefordert worden.
	X'40'	X'005B'	Beim 'ENAMP'-Aufruf zum Anlegen des Memory-Pools für die Access-Listen bzw. den IO-Area-Pool wurde nicht 'FIXED=YES' angegeben. Dies ist auch bei 'SCOPE=LOCAL' notwendig.

Funktion ENABLE IOAREA POOL

Es wird ein IO-Area-Pool eingerichtet oder der Aufrufer an einen bestehenden gekoppelt. In die Parameterliste wird eine Kennung eingetragen (FPAMIPID), die bei den folgenden OPEN Aufrufen zu benutzen ist. Wenn dabei dieselbe Parameterliste benutzt wird, ist die Kennung bereits eingetragen und muss daher nicht beachtet werden.
Von der Funktion ENAIPO werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet.

Format FCT=*ENAIPO

Operation	Operanden
FPAMSRV	[,FCT={*ENAIPO | adr | (r)}] [,IPONAME={'name' | adr | (r)}] [,IPOADDR=(elem1,elem2)] [,IPOSIZE={größe | adr | (r)}] [,RES={*NOT_SPECIFIED | *NO | *YES | adr | (r)}] MF=L
	MF=E,PARAM={adr | (r)}
	MF=D[,PREFIX={F | pre}]

(Teil 1 von 2)

Operation	Operanden
	MF={C | M}[,PREFIX={F | pre}][,MACID={PAM | macid}]

(Teil 2 von 2)

Operandenbeschreibung

FCT
Bestimmt die auszuführende FASTPAM-Funktion.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *ENAIPO
Ist die direkte Angabe der Funktion ENABLE IOAREA POOL.

Es wird ein IO-Area-Pool eingerichtet oder der Aufrufer an einen bestehenden IO-Area-Pool gekoppelt.
In die Parameterliste wird eine Kennung eingetragen (FPAMIPID), die bei den folgenden OPEN-Aufrufen zu benutzen ist. Wenn dabei dieselbe Parameterliste benutzt wird, ist die Kennung bereits eingetragen und braucht daher nicht beachtet zu werden.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die auszuführende Funktion ENABLE IOAREA POOL enthält.

= (r)
Ist ein Register, das den Wert für die Funktion ENABLE IOAREA POOL enthält.

IPOADDR
Bestimmt mittels einer Liste von 2 Elementen die Lage des IO-Area-Pools in einem Datenraum (Dataspace) bzw. im Programmraum (Programspace); (siehe Handbuch „Makroaufrufe an den Ablaufteil“ [2]).

= elem1
Enthält die auf 4 KByte ausgerichtete Anfangsadresse des Speicherbereichs im Datenraum bzw. Programmraum. Ist dieser Speicherbereich resident angelegt, darf er sich nicht mit dem Speicherbereich eines anderen IO-Area-Pools, mit dem Parameterlistenbereich eines Environments oder mit einem DIV-Fenster überlappen und muss vorher angefordert worden sein (REQM,REQMP...).

Liegt der Bereich in einem Memory-Pool, muss beim Anlegen des Memory-Pools im Makro ENAMP der Operand FIXED=YES angegeben worden sein.

Bei der Form MF=L ist nur die symbolische Adresse erlaubt, wobei symbolische Namen innerhalb einer DSECT ausgeschlossen sind, da ihre Adresse erst zur Laufzeit bekannt ist.

elem1 hat folgende Form:

$$\texttt{elem1} = \left\{ \begin{array}{l} \texttt{adr1} \\ \texttt{(r1)} \end{array} \right\}$$

adr1 Ist die symbolische Anfangsadresse des IO-Area-Pools im Daten- bzw. Programmraum.

(r1) Ist ein Register, das die Anfangsadresse des IO-Area-Pools im Daten- bzw. Programmraum enthält.

= elem2
Kennzeichnet den Adressraum.
elem2 = 0: der IO-Area-Pool liegt im Programmadressraum
elem2 ≠ 0: der IO-Area-Pool liegt in einem Datenraum mit dem ALET <elem2>
(ALET – Access List Entry Token; Zeiger auf einen Eintrag in der Zugriffsliste; siehe Handbuch „Makroaufrufe an den Ablaufteil“ [2]). Datenräume werden allerdings nur nicht-resident unterstützt.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

elem2 hat folgende Form:

$$\texttt{elem2} = \left\{ \begin{array}{l} \texttt{adr2} \\ \texttt{zahl} \\ \texttt{(r2)} \end{array} \right\}$$

zahl Ist der numerische Wert des ALETs bzw. 0.

adr2 Ist die symbolische Adresse eines 4 Byte langen Feldes, das den ALET bzw. 0 enthält (binär).

(r2) Ist ein Register, das den ALET bzw 0 enthält.

IPONAME
Bezeichnet den Namen des IO-Area-Pools.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= 'name'
Ist der Name des IO-Area-Pools.
Namenslänge: 1 ≤ 'name' ≤ 54 Zeichen.
Namensbildung:
1. Zeichen: Buchstabe oder Sonderzeichen #,@.
2. – 54. Zeichen: beliebige Kombination aus der Zeichenmenge (A,...,Z,0,...,9,$,#,@).
Das erste Blank (X'40') beendet den Namen.
Der Name muss in Hochkommata eingeschlossen werden.

= adr
Ist die symbolische Adresse eines 54 Byte langen Feldes, das den Namen des IO-Area-Pools enthält.

= (r)
Ist ein Register, das die Adresse eines 54 Byte langen Feldes mit dem Namen des IO-Area-Pools enthält.

IPOSIZE
Beschreibt die Größe des Speicherbereichs für den IO-Area-Pool in 4-KByte-Einheiten. Dieser Bereich muss zuvor in der entsprechenden Größe angelegt worden sein.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= größe
Größe des Speicherbereichs für den IO-Area-Pool in 4-KByte-Einheiten:
$(1 \leq \text{größe} \leq 2^{19})$.

= adr
Ist die symbolische Adresse eines 4 Byte langen Feldes, das die Größe des Speicherbereiches für den IO-Area-Pool in 4-KByte-Einheiten enthält.

= (r)
Ist ein Register, das die Größe des Speicherbereichs in 4-KByte-Einheiten enthält.

MACID
Zu MACID siehe Beschreibung beim Format FCT=*ENAENV, Seite 559.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

PARAM
Zu PARAM siehe Beschreibung beim Format FCT=*ENAENV, Seite 560.

PREFIX
Zu PREFIX siehe Beschreibung beim Format FCT=*ENAENV, Seite 560.

RES
Gibt an, ob der IO-Area-Pool resident oder nicht-resident angelegt werden soll.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *NOT_SPECIFIED
Der IO-Area-Pool besteht schon, und der Teilnehmer will sich daran anschließen, unabhängig davon, ob der IO-Area-Pool resident angelegt ist oder nicht.

= *YES
Der IO-Area-Pool soll resident angelegt werden.
In diesem Fall wird überprüft, ob die Benutzerkennung über die FASTPAM-Berechtigung verfügt und ob die beim Programmaufruf angeforderte Anzahl von residenten Seiten für den IO-Area-Pool ausreicht (Benutzerkatalog: „RESIDENT-PAGES“ bzw. „CLASSII“). Bei negativem Ergebnis erhält der Anwender die Fehleranzeige „FPAMNORE“ und der IO-Area-Pool wird nicht-resident angelegt.

Speicherplatz in Datenräumen wird nur nicht-resident zur Verfügung gestellt.

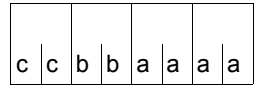
= *NO
Der IO-Area-Pool soll nicht resident angelegt werden.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für RES enthält.

= (r)
Ist ein Register, das den Wert für RES enthält.

*Mögliche Returncodes bei FCT=*ENAIPO*

Standard-header:



Über die Ausführung des Makros FPAMACC wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAIN-CODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Funktion erfolgreich ausgeführt.
	X'01'	X'0002'	Funktion nicht ausgeführt. Ungültiger Name des IO-Area-Pools
	X'01'	X'0007'	Funktion nicht ausgeführt. Ungültige Speicheradresse des IO-Area-Pools
	X'01'	X'0008'	Funktion nicht ausgeführt. Ungültige Größe des IO-Area-Pools
	X'01'	X'0012'	Funktion nicht ausgeführt. – Ungültige Angabe beim Operanden RES – Angabe NOT_SPECIFIED beim Operanden RES und der spezifizierte IO-Area-Pool existiert noch nicht
	X'20'	X'0028'	Funktion nicht ausgeführt. Systemfehler. Systemdiagnose einschalten.
	X'40'	X'0032'	Das Environment bzw. der IO-Area-Pool konnte nicht resident angelegt werden. Zu Testzwecken kann das Anwendersystem weiterarbeiten, allerdings auf Kosten der FASTPAM-Performancevorteile. Der subcode2 spezifiziert die Ursache.
X'01'	X'40'	X'0032'	Die USERID der Task, die das Environment bzw. den IO-Area-Pool eingerichtet hat, verfügt nicht über die FASTPAM-Berechtigung. Maßnahme: Systemverwalter verständigen.
X'02'	X'40'	X'0032'	Zu kleiner Realspeicher.
X'03'	X'40'	X'0032'	Das beim Programmstart angegebene Kontingent an residentem Hauptspeicher ist überschritten.
X'04'	X'40'	X'0032'	Konnektierung an ein nicht-residentes Environment bzw. an einen nicht-residenten IO-Area-Pool.
X'05'	X'40'	X'0032'	In vorliegender FASTPAM-Version werden Datenräume (Data-Spaces) nur nicht-resident unterstützt.
	X'40'	X'0035'	Während der Ausführung der Funktionen *ENAENV/*ENAIPO läuft eine Ein-/Ausgabe auf die zu fixierenden Speicherseiten. Dieser Returncode kommt nur beim Einrichten des Environments bzw. IO-Area-Pools. Während der Konnektierung dürfen Ein-/Ausgaben laufen.

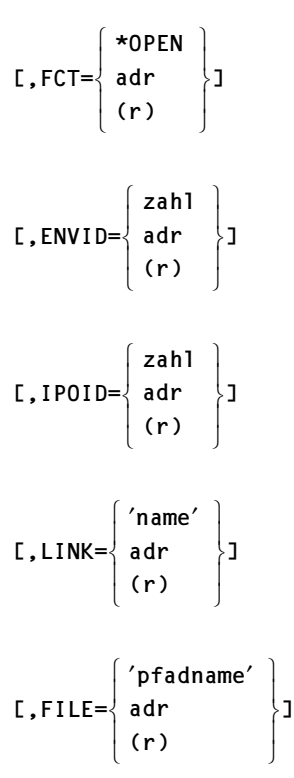
X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'0037'	Systembetriebsmittelengpass. Maßnahme: Systemverwalter verständigen.
	X'40'	X'0038'	An das genannte FASTPAM-Betriebsmittel können sich nur TPR-Tasks konnektieren.
	X'40'	X'0039'	An das genannte FASTPAM-Betriebsmittel können sich nur Tasks einer Kennung mit FASTPAM-Privileg konnektieren.
	X'40'	X'003D'	Der Teilnehmer will sich an einen bestehenden IO-Area-Pool mit einem anderen Operandenwert für IPOADDR konnektieren.
	X'40'	X'003E'	Der Teilnehmer will sich an einen bestehenden IO-Area-Pool mit einem anderen Operandenwert für IPOSIZE konnektieren.
	X'40'	X'0049'	Die Task ist bereits an den IO-Area-Pool konnektiert.
	X'40'	X'004A'	Der angegebene Anwenderspeicher überlappt sich mit einem DIV-Fenster. Dieser Returncode kommt nur im residenten Fall.
	X'40'	X'004B'	Der angegebene Anwenderspeicher überlappt sich mit einem bereits verwendeten FASTPAM-Anwenderspeicher. Dieser Returncode wird nur ausgegeben, wenn mit residentem Environment bzw. mit residentem IO-Area-Pool gearbeitet wird.
	X'40'	X'0057'	Der Speicher für den IO-Area-Pool ist nicht vollständig angefordert worden.
	X'40'	X'005B'	Beim 'ENAMP'-Aufruf zum Anlegen des Memory-Pools für die Access-Listen bzw. den IO-Area-Pool wurde nicht 'FIXED=YES' angegeben. Dies ist auch bei 'SCOPE=LOCAL' notwendig.

Funktion OPEN

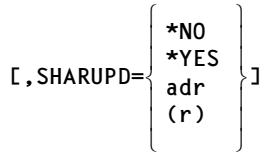
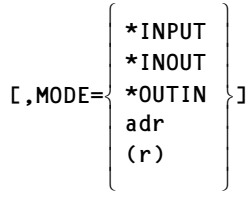
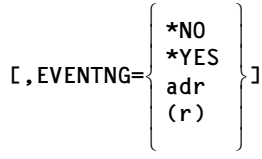
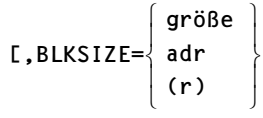
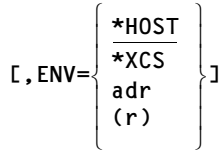
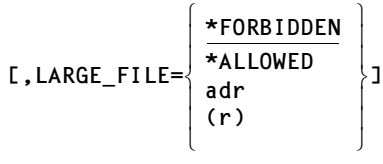
Mit dieser Funktion kann mit den beim ENABLE ENVIRONMENT und ENABLE IOAREA POOL erhaltenen Kurzkennungen eine PAM Datei eröffnet werden.
In die Parameterliste wird eine Kurzkennung eingetragen (FPAMOPID), die in die FPAMACC Parameterlisten für die folgenden Dateizugriffe zu übertragen ist.

Von der Funktion OPEN werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet.

Format FCT=*OPEN

Operation	Operanden
FPAMSRV	

(Teil 1 von 3)

Operation	Operanden
	      MF=L
	MF=E,PARAM=(adr | (r))

(Teil 2 von 3)

Operation	Operanden
	MF=D[,PREFIX={F / pre}]
	MF={C / M}[,PREFIX={F / pre}][,MACID={PAM / macid}]

(Teil 3 von 3)

Operandenbeschreibung

BLKSIZE
Bestimmt die Blockgröße der nachfolgenden Ein-/Ausgaben in 4-KByte-Einheiten.
Der Wert für BLKSIZE darf den Wert, der bei der Funktion ENABLE ENVIRONMENT beim Operanden MAXIOLN angegeben wurde, nicht übersteigen.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= größe
Gibt die Blockgröße in 4-KByte-Einheiten an: 1 ≤ größe ≤ 8

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das die Blockgröße in 4-KByte-Einheiten enthält (binär).

= (r)
Ist ein Register, das die Blockgröße in 4-KByte-Einheiten enthält.

ENV
Beeinflusst die Verträglichkeit paralleler Eröffner in Abhängigkeit von ihrem Ablaufort (vgl. dazu „Verträglichkeits-Matrix FASTPAM mit UPAM/FASTPAM/DIV“ auf Seite 69).

= *HOST
Die maximal mögliche Parallelität ist auf Eröffner beschränkt, die im gleichen Host ablaufen.

= *XCS
Die Eröffner können in verschiedenen Hosts eines XCS-Rechnerverbunds ablaufen, ohne dass dadurch die Verträglichkeit eingeschränkt wird (z.B. können Schreiboperationen mit SHARUPD=*YES parallel ablaufen).

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für ENV enthält.

= (r)
Ist ein Register, das den Wert für ENV enthält.

ENVID
Bezeichnet die Kurzkennung des Environments, mit dem die Datei eröffnet werden soll. Wird dieselbe Parameterliste benutzt wie beim ENABLE ENVIRONMENT, ist die Angabe nicht erforderlich, da sich die Kurzkennung bereits in der Parameterliste befindet (FPAMENID).

Bei der Form MF=L ist die ENVID-Angabe nicht möglich.

= zahl
Direkte Angabe der Kurzkennung als dezimaler numerischer Wert.

= adr
Ist die Adresse eines 4 Byte langen Feldes, das die Kurzkennung enthält.

= (r)
Ist ein Register, das die Kurzkennung enthält.

EVENTNG
Legt fest, ob der Anwender bei den asynchron ausgeführten Dateizugriffen das Auftragsende über Eventing mitgeteilt bekommt (vgl. dazu Abschnitt „FASTPAM-Funktionen, Eventing Verarbeitung“, Handbuch „Einführung in das DVS“ [1]).

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *YES
Der Anwender will mit Eventing arbeiten. Dies wird nur akzeptiert, wenn beim Einrichten des Environments auch EVENTING=*YES angegeben worden ist.

= *NO
Der Anwender will nicht mit Eventing arbeiten.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für EVENTNG enthält.

= (r)
Ist ein Register, das den Wert für EVENTNG enthält.

FCT
Bestimmt die auszuführende FASTPAM-Funktion.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *OPEN
Direkte Angabe der Funktion OPEN.
Mit dieser Funktion kann mit den beim ENABLE ENVIRONMENT und ENABLE IOAREA POOL erhaltenen Kurzkennungen eine PAM-Datei eröffnet werden.
In die Parameterliste wird eine Kurzkennung eingetragen (FASTOPID), die in die FPAMACC Parameterlisten für die folgenden Dateizugriffe zu übertragen ist.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die die Funktion OPEN enthält.

= (r)
Ist ein Register, das den Wert für die Funktion OPEN enthält.

FILE
Gibt den Pfadnamen der Datei an. Falls eine Angabe beim Operanden LINK erfolgte, wird eine FILE-Angabe nicht ausgewertet.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= '*pfadname*'
<c-string 1..54: filename 1..54>
Der Name muss in Hochkommata eingeschlossen werden.

= adr
Ist die Adresse eines 54 Byte langen Feldes, das den Pfadnamen enthält.

= (r)
Ist ein Register, das die Adresse des mit dem Pfadnamen enthält.

IPOID
Bezeichnet die Kurzkennung des IO-Area-Pools, mit dem die Datei eröffnet werden soll. Wird dieselbe Parameterliste benutzt wie beim ENABLE IOAREA POOL, ist die Angabe nicht erforderlich, da sich die Kurzkennung bereits in der Parameterliste befindet.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= zahl
Ist die Kurzkennung des IO-Area-Pools als dezimaler numerischer Wert.

= adr
Ist die Adresse eines 4 Byte langen Feldes, das die Kurzkennung enthält.

= (r)
Ist ein Register, das die Kurzkennung enthält.

LARGE_FILE
Gibt an, ob die zu eröffnende Datei eine „große Datei“ werden darf, also eine Datei mit einer Dateigröße ≥ 32 GB.

Voreinstellung: LARGE_FILE = *FORBIDDEN

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *FORBIDDEN
Die Datei darf keine „große Datei“ werden.

= *ALLOWED
Die Datei darf eine „große Datei“ werden.

= adr
Ist die Adresse eines 8 Byte langen Feldes, das den Wert für LARGE-FILE enthält.

= (r)
Ist ein Register, das die Adresse eines 8 Byte langen Feldes mit dem Wert für LARGE-FILE enthält.

LINK
Spezifiziert den Dateikettungsnamen.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= 'name'
Dateikettungsname mit: <c-string 1..8> (eingeschlossen in Hochkommata)

Soll der Dateikettungsname über die Kommandoschnittstelle ansprechbar sein, muss er dem Datentyp <structured_name 1..8> entsprechen (siehe Handbuch „Kommandos“ [3]).

= adr
Ist die Adresse eines 8 Byte langen Feldes, das den Dateikettungsnamen enthält.

= (r)
Ist ein Register, das die Adresse des Feldes mit dem Dateikettungsnamen enthält.

MACID
Zu MACID siehe Beschreibung beim Format FCT=*ENAENV, Seite 559.

MF
Die Formen des MF-Operanden sind detailliert im Anhang, Seite 870 beschrieben.

MODE
Bestimmt den OPEN-Modus (vgl. dazu Abschnitt „Multi-User-Betrieb an einem Rechner“ auf Seite 68 sowie Abschnitt „FASTPAM-Funktionen, Multi-User-Betrieb...“ im Handbuch „Einführung in das DVS“ [1]).

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *INPUT
Die Datei wird nur gelesen.
Unabhängig vom SHARUPD-Modus sind parallele INPUT-Eröffnungen möglich, auch von den Zugriffsmethoden UPAM und DIV.
Die Datei muss existieren, d.h., sie muss schon einmal mit OUTIN eröffnet worden sein.

= *INOUT
In die Datei kann auch geschrieben werden.
Ob parallele Eröffnungen möglich sind, hängt vom SHARUPD-Modus und vom Operanden ENV ab.
Die Datei muss existieren, d.h., sie muss schon einmal mit OUTIN eröffnet worden sein.

= *OUTIN
In die Datei kann auch geschrieben werden.
Die Datei wird jedoch neu erstellt, d.h. nach dem OPEN ist die Datei leer.
Ob parallele Eröffnungen möglich sind, hängt vom SHARUPD-Modus und vom Operanden ENV ab.
Bei einem Multi-User Betrieb (SHARUPD=*YES) muss ein MODE=*OUTIN-Eröffner immer der Erste sein, sonst wird er zurückgewiesen.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für den OPEN-Modus enthält.

= (r)
Ist ein Register, das den Wert für den OPEN-Modus enthält.

PARAM
Zu PARAM siehe Beschreibung beim Format FCT=*ENAENV, Seite 560.

PREFIX
Zu PREFIX siehe Beschreibung beim Format FCT=*ENAENV, Seite 560.

SHARUPD
Steuert den Multi-User-Betrieb (vgl. dazu Abschnitt „FASTPAM-Funktionen, Multi-User-Betrieb...“, Handbuch „Einführung in das DVS“ [1]).

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *NO
Es sind mehrere parallele Leser (MODE=*INPUT) oder genau ein Schreiber (MODE=*INOUT | *OUTIN) erlaubt.

= *YES
Es sind mehrere parallele Schreiber und Leser erlaubt.

Hinweise

FASTPAM bietet keine Funktion „Sperren von Blöcken“ (LOCK/UNLOCK-Funktionen). Der Anwender muss einen Sperrmechanismus selbst zur Verfügung stellen. Parallel zugreifende UPAM-Anwendungen müssen anders synchronisiert werden.

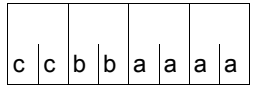
Bei jedem Aufruf des Allocators wird die Dateigröße überprüft.
Wenn bei dieser Überprüfung eine Dateigröße ≥ 32 GB ermittelt wird und im zugehörigen FCB das Attribut LARGE_FILE=*FORBIDDEN bzw. in der TFT das Attribut EXCEED-32GB=*FORBIDDEN gesetzt ist, wird die Verarbeitung abgebrochen.
FASTPAM liefert in diesem Fall den Returncode `X'00400145'` in seiner eigenen Parameterliste FPAMACC(I).

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für SHARUPD enthält.

= (r)
Ist ein Register, das den Wert für SHARUPD enthält.

*Mögliche Returncodes bei FCT=*OPEN*

Standard-header:



Über die Ausführung des Makros FPAMSRV wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Funktion erfolgreich ausgeführt.
	X'01'	X'0009'	Funktion nicht ausgeführt. Ungültige Angabe bei SHARE UPDATE max. 4K bzw. 32k
	X'01'	X'000B'	Funktion nicht ausgeführt. Ungültiges Eventing
	X'01'	X'000C'	Funktion nicht ausgeführt. Ungültige Angabe bei MODE

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'01'	X'000E'	Funktion nicht ausgeführt. Ungültige Angabe für logische Blocklänge (BLKSIZE) *Hinweis* Beim Kommando ADD-FILE-LINK wird BLKSIZE in 2-KByte-Einheiten angegeben; der hierbei angegebene Wert entspricht daher einer halb so großen BLKSIZE-Angabe beim FASTPAM-OPEN.
	X'01'	X'000F'	Funktion nicht ausgeführt. Ungültige Kurzkennung des Environments
	X'01'	X'0010'	Funktion nicht ausgeführt. Ungültige Kurzkennung des IO-Area-Pools
	X'01'	X'0014'	Funktion nicht ausgeführt. Ungültige Angabe bei ENV
	X'01'	X'0015'	Funktion nicht ausgeführt. Ungültige Angabe bei LARGE_FILE
	X'20'	X'0028'	Funktion nicht ausgeführt. Systemfehler. Systemdiagnose einschalten.
	X'40'	X'0033'	Fehler beim allgemeinen DMS OPEN/CLOSE. Der DMS-Returncode wird im Feld FPAMDMSC zurückgeliefert. Maßnahme: DMS-Returncode auswerten.
	X'40'	X'0037'	Systembetriebsmittelengpass. Maßnahme: Systemverwalter verständigen.
	X'40'	X'0042'	Der beim FASTPAM-OPEN angegebene Wert für BLKSIZE stimmt nicht mit dem im Katalogeintrag überein.
	X'40'	X'004D'	Die genannte Datei ist keine PAM-Datei.
	X'40'	X'0050'	RFA wird von FASTPAM nicht unterstützt.
	X'40'	X'0051'	SPD wird von FASTPAM nicht unterstützt.
	X'40'	X'0052'	PPD wird von FASTPAM nicht unterstützt.
	X'40'	X'0053'	Banddateien werden von FASTPAM nicht unterstützt.
	X'40'	X'0054'	*DUMMY wird von FASTPAM nicht unterstützt.
	X'40'	X'0055'	Die Sekundärallokierung reicht nicht aus, um einen logischen Block aufzunehmen.
	X'40'	X'0056'	Ein nicht-privilegierter Anwender gibt die Kurzkennung eines Environments oder IO-Area-Pools an, der von einem privilegierten Anwender angelegt wurde.
	X'40'	X'0058'	FASTPAM unterstützt nur Dateien mit dem Attribut 'BLKCTRL = NO'.
	X'40'	X'005A'	Beim FILE-Aufruf wurde 'WRCHK=YES' angegeben.

Funktion CLOSE

Mit dieser Funktion kann mit der beim OPEN erhaltenen Kurzkennung (OPENID) die PAM-Datei wieder geschlossen werden.

Von der Funktion CLOSE werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet.

Format FCT=*CLOSE

Operation	Operanden
FPAMSRV	[,FCT={*CLOSE | adr | (r)}] [,OPENID={zahl | adr | (r)}] [,LASTBLK={zahl | adr | (r)}] MF=L
	MF=E,PARAM={adr | (r)}
	MF=D[,PREFIX={F | pre}]
	MF={C | M}[,PREFIX={F | pre}][,MACID={PAM | macid}]

Operandenbeschreibung

FCT
Bestimmt die auszuführende FASTPAM-Funktion.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *CLOSE
Direkte Angabe der Funktion „Schließen der Datei“.

Mit dieser Funktion kann mit der beim OPEN erhaltenen Kurzkennung (OPENID) die PAM Datei wieder geschlossen werden.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die Funktion CLOSE enthält.

= (r)
Ist ein Register, das den Wert für die Funktion CLOSE enthält.

LASTBLK
Mit diesem Parameter kann der Anwender den letzten logischen Block der Datei explizit setzen, wenn er die Datei mit MODE=*INOUT/*OUTIN und SHARUPD=*NO geöffnet hat. Der angegebene Block muss innerhalb der Datei liegen.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= zahl
Ist die direkte Angabe eines dezimalen numerischen Wertes für den letzten 4-KByte-Block der Datei.

= adr
Ist die symbolische Adresse eines 4 Byte langen Feldes, das den numerischen Wert (binär) für den letzten logischen Block der Datei enthält.

= (r)
Ist ein Register, das den numerischen Wert für LASTBLK enthält.

MACID
Zu MACID siehe Beschreibung beim Format FCT=*ENAENV, Seite 559.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

PARAM
Zu PARAM siehe Beschreibung beim Format FCT=*ENAENV, Seite 560.

PREFIX
Zu PREFIX siehe Beschreibung beim Format FCT=*ENAENV, Seite 560.

OPENID
Bezeichnet die Kurzkennung des OPEN, für den die CLOSE-Funktion ausgeführt werden soll.
Wird dieselbe Parameterliste benutzt wie beim OPEN, ist die Angabe nicht erforderlich, da sich die Kurzkennung bereits in der Parameterliste (Feld FPAMOPID) befindet.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

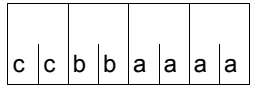
= zahl
Ist die direkte Angabe eines dezimalen numerischen Wertes für OPENID.

= adr
Ist die Adresse eines 4 Byte langen Feldes, das die Kurzkennung enthält.

= (r)
Ist ein Register, das die Kurzkennung enthält.

*Mögliche Returncodes bei FCT=*CLOSE*

Standard-header:



Über die Ausführung des Makros FPAMSRV wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAIN-CODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Funktion erfolgreich ausgeführt.
	X'01'	X'0011'	Funktion nicht ausgeführt. Ungültige Kurzkennung des OPEN
	X'01'	X'0013'	Funktion nicht ausgeführt. Ungültige Angabe für den letzten Block. Die Operation CLOSE wurde bis auf die Veränderung des Last Page Pointers ausgeführt.
	X'20'	X'0028'	Funktion nicht ausgeführt. Systemfehler. Systemdiagnose einschalten.
	X'40'	X'0033'	Fehler beim allgemeinen DMS OPEN/CLOSE. Der DMS-Returncode wird im Feld FPAMDMSC zurückgeliefert. Maßnahme: DMS-Returncode auswerten.
	X'40'	X'004E'	Beim FPAMSRV-Aufruf mit der Funktion *CLOSE wurde der Operand LASTBLK angegeben. Dieser wird ignoriert, da die Datei mit MODE=*INPUT oder mit SHARUPD=*YES eröffnet wurde.
	X'40'	X'0059'	Eine TU-Task benutzt eine TPR-Open-Id.

Funktion DISABLE IOAREA POOL

Es wird die Verbindung des Teilnehmers zum IO-Area-Pool gelöst. Der IO-Area-Pool wird abgebaut, wenn der Aufrufer der letzte Teilnehmer ist. Der IO-Area-Pool wird über die beim ENABLE IOAREA POOL erhaltene Kurzkennung angesprochen.

Von der Funktion DISIPO werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet.

Format FCT=*DISIPO

Operation	Operanden
FPAMSRV	[,FCT={*DISIPO / adr / (r)}] [,IPOID={zahl / adr / (r)}] MF=L
	MF=E,PARAM={adr / (r)}
	MF=D[,PREFIX={F / pre}]
	MF={C / M}[,PREFIX={F / pre}][,MACID={PAM / macid}]

Operandenbeschreibung

FCT
Bestimmt die auszuführende FASTPAM-Funktion.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *DISIPO
Ist die direkte Angabe der Funktion DISABLE IOAREA POOL.

Es wird die Verbindung des Teilnehmers zum IO-Area-Pool gelöst. Der IO-Area-Pool wird abgebaut, wenn der Aufrufer der letzte Teilnehmer ist.
Der IO-Area-Pool wird über die beim ENABLE IOAREA POOL erhaltene Kurzkennung angesprochen.

Die Funktion wird nicht ausgeführt, wenn noch Dateien mit dem IO-Area-Pool geöffnet sind (Returncode FPAMOFI).

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die Funktion DISABLE IOAREA POOL enthält.

= (r)
Ist ein Register, das den Wert für die Funktion DISABLE IOAREA POOL enthält.

IPOID
Bezeichnet die Kurzkennung des IO-Area-Pools, zu dem die Verbindung gelöst werden soll bzw. der abgebaut werden soll.
Wird dieselbe Parameterliste benutzt wie beim ENABLE IOAREA POOL, ist die Angabe nicht erforderlich, da sich die Kurzkennung bereits in der Parameterliste befindet (Feld FPAMIPID).

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= zahl
Ist die direkte Angabe eines dezimalen numerischen Wertes für die Kurzkennung des IO-Area-Pools.

= adr
Ist die Adresse eines 4 Byte langen Feldes, das die Kennung für den IO-Area-Pool enthält.

= (r)
Ist ein Register, das die Kurzkennung für den IO-Area-Pool enthält.

MACID
Zu MACID siehe Beschreibung beim Format FCT=*ENAENV, Seite 559.

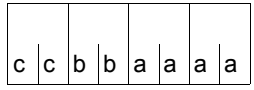
MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

PARAM
Zu PARAM siehe Beschreibung beim Format FCT=*ENAENV, Seite 560.

PREFIX
Zu PREFIX siehe Beschreibung beim Format FCT=*ENAENV, Seite 560.

*Mögliche Returncodes bei FCT=*DISIPO*

Standard-header:



Über die Ausführung des Makros FPAMSRV wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAIN-CODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Funktion erfolgreich ausgeführt.
	X'01'	X'0010'	Funktion nicht ausgeführt. Ungültige Kurzkennung des IO-Area-Pools
	X'20'	X'0028'	Funktion nicht ausgeführt. Systemfehler. Systemdiagnose einschalten
	X'40'	X'0036'	Während der Ausführung der Funktionen *DISIPO/*DISENV sind noch Dateien mit dem entsprechenden Environment bzw. IO-Area-Pool geöffnet.
	X'40'	X'0056'	Ein TU-Anwender will einen aus TPR angelegten FASTPAM-IO-Area-Pool freigeben.

Funktion DISABLE ENVIRONMENT

Es wird die Verbindung des Teilnehmers zum Environment gelöst. Das Environment wird abgebaut, wenn der Aufrufer der letzte Teilnehmer ist. Das Environment wird über die beim ENABLE ENVIRONMENT erhaltene Kurzkennung angesprochen.
Von der Funktion DISENV werden nur die nachfolgend beschriebenen Funktionsoperanden ausgewertet.

Format FCT=*DISENV

Operation	Operanden
FPAMSRV	[,FCT={ *DISENV / adr / (r) }] [,ENVID={ zahl / adr / (r) }] MF=L
	MF=E,PARAM={ adr / (r) }
	MF=D[,PREFIX={ F / pre }]
	MF={ C / M }[,PREFIX={ F / pre }][,MACID={ PAM / macid }]

Operandenbeschreibung

ENVID
Bezeichnet die Kurzkennung des Environments, zu dem die Verbindung gelöst werden soll bzw. das abgebaut werden soll. Wird dieselbe Parameterliste benutzt wie beim ENABLE ENVIRONMENT, ist die Angabe nicht erforderlich, da sich die Kurzkennung bereits in der Parameterliste befindet (FPAMENID).

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= zahl
Ist die direkte Angabe eines dezimalen numerischen Wertes für die Kurzkennung des Environments.

= adr
Ist die Adresse eines 4 Byte langen Feldes, das die Kurzkennung des Environments enthält.

= (r)
Ist ein Register, das die Kurzkennung des Environments enthält.

FCT
Bestimmt die auszuführende FASTPAM-Funktion.

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= *DISENV
Direkte Angabe der Funktion DISABLE ENVIRONMENT.

Es wird die Verbindung des Teilnehmers zum Environment gelöst. Das Environment wird abgebaut, wenn der Aufrufer der letzte Teilnehmer ist.
Das Environment wird über die beim ENABLE ENVIRONMENT erhaltene Kurzkennung angesprochen.

Die Funktion wird nicht ausgeführt, wenn noch Dateien mit dem Environment geöffnet sind (Returncode FPAMOFI).

Bei der Form MF=L ist nur die direkte Angabe erlaubt.

= adr
Ist die symbolische Adresse eines 1 Byte langen Feldes, das den Wert für die Funktion DISABLE ENVIRONMENT enthält.

= (r)
Ist ein Register, das den Wert für die Funktion DISABLE ENVIRONMENT enthält.

MACID
Zu MACID siehe Beschreibung beim Format FCT=*ENAENV, Seite 559.

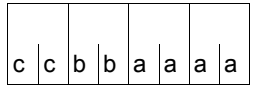
MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

PARAM
Zu PARAM siehe Beschreibung beim Format FCT=*ENAENV, Seite 560.

PREFIX
Zu PREFIX siehe Beschreibung beim Format FCT=*ENAENV, Seite 560.

*Mögliche Returncodes bei FCT=*DISENV*

Standard-header:



Über die Ausführung des Makros FPAMSRV wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAIN-CODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Funktion erfolgreich ausgeführt.
	X'01'	X'000F'	Funktion nicht ausgeführt. Ungültige Kurzkennung des Environments
	X'20'	X'0028'	Funktion nicht ausgeführt. Systemfehler. Systemdiagnose einschalten.
	X'40'	X'0036'	Während der Ausführung der Funktionen *DISIPO/*DISENV sind noch Dateien mit dem entsprechenden Environment bzw. IO-Area-Pool geöffnet.
	X'40'	X'0056'	Ein TU-Anwender will ein aus TPR angelegtes FASTPAM-Environment freigeben.

Returncodes des Makros FPAMSRV

Die mit der C- oder D-Form des Makros generierten Feldnamen und EQU-Anweisungen für die Returncodes beginnen standardmäßig mit der Zeichenfolge FPAM, die durch PREFIX und MACID geändert werden kann.

Die Returncodes werden im Header der Parameterliste (Standardheader) hinterlegt:

– Der Main Returncode in einem Halbwort mit dem Namen FPAMMRET.
– Der Subcode1 in einem Byte mit dem Namen FPAMSR1.
  Subcode1 beschreibt Fehlerklassen, die es dem Aufrufer ermöglichen, auf ähnliche Fehlerfälle gleich zu reagieren.
  Der Aufrufer kann sich sowohl am Maincode als auch am Subcode1 orientieren.
– Der Subcode2 in einem Byte mit dem Namen FPAMSR2.
  Der Subcode2 spezifiert einzelne Maincodes genauer.

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

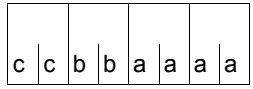
Falls Returncodes nicht im Header abgelegt werden können (z.B. wenn er nicht zugreifbar ist), wird das aufrufende Programm mit einer Fehlermeldung beendet und das STXIT-Ereignis „nicht behebbarer Programmfehler“ erzeugt.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Im Folgenden werden die Mainreturncodes den entsprechenden Subcode1 Klassen zugeordnet und mittels Subcode2 genauer beschrieben.

Standard-header:



Über die Ausführung des Makros FPAMSRV wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAIN-CODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Funktion erfolgreich ausgeführt.
	X'01'	X'0001'	Funktion nicht ausgeführt. Ungültiger Name des Environments
	X'01'	X'0002'	Funktion nicht ausgeführt. Ungültiger Name des IO-Area-Pools;
	X'01'	X'0005'	Funktion nicht ausgeführt. Ungültige Adresse der Operandenliste
	X'01'	X'0006'	Funktion nicht ausgeführt. Ungültige Anzahl von Operandenlisten
	X'01'	X'0007'	Funktion nicht ausgeführt. Ungültige Speicheradresse des IO-Area-Pools
	X'01'	X'0008'	Funktion nicht ausgeführt. Ungültige Größe des IO-Area-Pools
	X'01'	X'0009'	Funktion nicht ausgeführt. Ungültige Angabe bei SHARE UPDATE
	X'01'	X'000A'	Funktion nicht ausgeführt. Ungültige Angabe bei MAXIOLN
	X'01'	X'000B'	Funktion nicht ausgeführt. Ungültige Angabe bei EVENTNG
	X'01'	X'000C'	Funktion nicht ausgeführt. Ungültige Angabe bei MODE
	X'01'	X'000D'	Funktion nicht ausgeführt. Ungültige Ereigniskurzkennung
	X'01'	X'000E'	Funktion nicht ausgeführt. Ungültige Angabe für logische Blocklänge (BLKSIZE) *Hinweis* Beim Kommando ADD-FILE-LINK wird BLKSIZE in 2K-Einheiten angegeben; der hierbei angegebene Wert entspricht daher einer halb so großen BLKSIZE-Angabe beim FASTPAM-OPEN.
	X'01'	X'000F'	Funktion nicht ausgeführt. Ungültige Kurzkennung des Environments

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'01'	X'0010'	Funktion nicht ausgeführt. Ungültige Kurzkennung des IO-Area-Pools
	X'01'	X'0011'	Funktion nicht ausgeführt. Ungültige Kurzkennung des OPEN
	X'01'	X'0012'	Funktion nicht ausgeführt. – Ungültige Angabe beim Operanden RES – Angabe NOT_SPECIFIED beim Operanden RES und das angegebene Environment und der angegebene IO-Area-Pool existieren noch nicht
	X'01'	X'0013'	Ungültige Angabe für den letzten Block. Die Operation CLOSE wurde bis auf die Veränderung des Last Page Pointers ausgeführt.
	X'01'	X'0014'	Funktion nicht ausgeführt. Ungültige Angabe bei ENV
	X'01'	X'0015'	Funktion nicht ausgeführt. Ungültige Angabe bei LARGE_FILE
	X'02'		Funktion nicht ausgeführt. Angegebene Funktion ist nicht verfügbar.
	X'03'		Funktion nicht ausgeführt. Angegebene Schnittstellen-Version wird nicht unterstützt.
	X'20'	X'0028'	Funktion nicht ausgeführt. Systemfehler. Systemdiagnose einschalten.
	X'40'	X'0032'	Das Environment bzw. der IO-Area-Pool konnte nicht resident angelegt werden. Zu Testzwecken kann das Anwendersystem weiterarbeiten, allerdings auf Kosten der FASTPAM-Performancevorteile. Der subcode2 spezifiziert die Ursache.
X'01'	X'40'	X'0032'	Die USERID der Task, die das Environment bzw. den IO-Area-Pool eingerichtet hat, verfügt nicht über die FASTPAM-Berechtigung. Maßnahme: Systemverwalter verständigen.
X'02'	X'40'	X'0032'	Zu kleiner Realspeicher.
X'03'	X'40'	X'0032'	Das beim Programmstart angegebene Kontingent an residentem Hauptspeicher ist überschritten.
X'04'	X'40'	X'0032'	Konnektierung an ein nicht-residentes Environment, bzw. an einen nicht-residenten IO-Area-Pool.
X'05'	X'40'	X'0032'	In vorliegender FASTPAM-Version werden Datenräume (Data-Spaces) nur nicht-resident unterstützt.

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'0033'	Fehler beim allgemeinen DMS OPEN/CLOSE. Der DMS-Returncode wird im Feld FPAMDMSC zurückgeliefert. Maßnahme: DMS-Returncode auswerten.
	X'40'	X'0035'	Während der Ausführung der Funktionen *ENAENV/*ENAIPO läuft eine Ein-/Ausgabe auf die zu fixierenden Speicherseiten. Dieser Returncode kommt nur beim Einrichten des Environments bzw. IO-Area-Pools. Während der Konnektierung dürfen Ein-/Ausgaben laufen.
	X'40'	X'0036'	Während der Ausführung der Funktionen *DISIPO/*DISENV sind noch Dateien mit dem entsprechenden Environment bzw. IO-Area-Pool geöffnet.
	X'40'	X'0037'	Systembetriebsmittelengpass. Maßnahme: Systemverwalter verständigen.
	X'40'	X'0038'	An das genannte FASTPAM-Betriebsmittel können sich nur TPR-Tasks konnektieren.
	X'40'	X'0039'	An das genannte FASTPAM-Betriebsmittel können sich nur Tasks einer Kennung mit FASTPAM-Privileg konnektieren.
	X'40'	X'003B'	Der Teilnehmer will sich an ein bestehendes Environment mit einem anderen Operandenwert für ACCLSTS konnektieren.
	X'40'	X'003C'	Der Teilnehmer will sich an ein bestehendes Environment mit einem anderen Operandenwert für ACCNUMB konnektieren.
	X'40'	X'003D'	Der Teilnehmer will sich an einen bestehenden IO-Area-Pool mit einem anderen Operandenwert für IPOADDR konnektieren.
	X'40'	X'003E'	Der Teilnehmer will sich an einen bestehenden IO-Area-Pool mit einem anderen Operandenwert für IPOSIZE konnektieren.
	X'40'	X'003F'	Der Teilnehmer will sich an ein bestehendes Environment mit einem anderen Operandenwert für MAXIOLN konnektieren.
	X'40'	X'0040'	Der Teilnehmer will sich an ein bestehendes Enviroment mit einem anderen Operandenwert für EVENTNG konnektieren.
	X'40'	X'0041'	Der Teilnehmer will sich an ein bestehendes Environment mit einer anderen Ereigniskennung konnektieren.
	X'40'	X'0042'	Der beim FASTPAM-OPEN angegebene Wert für BLKSIZE stimmt nicht mit dem im Katalogeintrag überein.
	X'40'	X'0046'	Der Anwender will sich an ein bestehendes Environment anschließen, ist aber nicht an die zugehörige Ereigniskurzkennung angeschlossen.
	X'40'	X'0047'	Der Gültigkeitsbereich der Ereigniskennung ist kleiner als der Gültigkeitsbereich des FPAMACC-Parameterlistenspeichers.
	X'40'	X'0048'	Die Task ist an das Environment bereits konnektiert.

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'0049'	Die Task ist an den IO-Area-Pool bereits konnektiert.
	X'40'	X'004A'	Der angegebene Anwenderspeicher überlappt sich mit einem DIV-Fenster. Dieser Returncode kommt nur im residenten Fall.
	X'40'	X'004B'	Der angegebene Anwenderspeicher überlappt sich mit einem bereits verwendeten FASTPAM-Anwenderspeicher. Dieser Returncode wird nur ausgegeben, wenn mit residentem Environment bzw. mit residentem IO-Area-Pool gearbeitet wird.
	X'40'	X'004C'	Der Speicher für die FPAMACC-Parameterlisten ist nicht vollständig angefordert worden.
	X'40'	X'004D'	Die genannte Datei ist keine PAM-Datei.
	X'40'	X'004E'	Beim FPAMSRV-Aufruf mit der Funktion *CLOSE wurde der Operand LASTBLK angegeben. Dieser wird ignoriert, da die Datei mit MODE=*INPUT oder mit SHARUPD=*YES eröffnet wurde.
	X'40'	X'0050'	RFA wird von FASTPAM nicht unterstützt.
	X'40'	X'0051'	SPD wird von FASTPAM nicht unterstützt.
	X'40'	X'0052'	PPD wird von FASTPAM nicht unterstützt.
	X'40'	X'0053'	Banddateien werden von FASTPAM nicht unterstützt.
	X'40'	X'0054'	*DUMMY wird von FASTPAM nicht unterstützt.
	X'40'	X'0055'	Die angegebene Sekundärallokierung reicht nicht aus, um einen logischen Block aufzunehmen.
	X'40'	X'0056'	Ein TU-Task will mit einem aus TPR angelegten FASTPAM-Betriebsmittel eine Datei eröffnen oder dieses freigeben.
	X'40'	X'0057'	Der Speicher für den IO-Area-Pool ist nicht vollständig angefordert worden.
	X'40'	X'0058'	FASTPAM unterstützt nur Dateien mit dem Attribut 'BLKCTRL=NO'.
	X'40'	X'0059'	Ein TU-Task benutzt eine TPR-Open-Id.
	X'40'	X'005A'	Beim FILE-Aufruf wurde 'WRCHK=YES' angegeben.
	X'40'	X'005B'	Beim 'ENAMP'-Aufruf zum Anlegen des Memory-Pools für die Access-Listen bzw. den IO-Area-Pool wurde nicht 'FIXED=YES' angegeben. Dies ist auch bei 'SCOPE=LOCAL' notwendig.

Hinweise zu den Returncodes

Ursache für Returncodes mit Subcode1 = X'01' (PARAMETER ERROR):

– der jeweilige Parameter wurde falsch angegeben
– bei Operandenwerten, die durch ADD-FILE-LINK überschrieben werden können, ist der durch dieses Kommando eingestellte Operandenwert unzulässig. Zulässig sind nur die Werte, die auch beim OPEN angegeben werden können.

Wird ein Fehler festgestellt, der den gesamten Bereich der FPAMACC-Parameterlisten oder des IO-Area-Pools betrifft (Bsp: der IO-Area-Pool liegt nur zum Teil in einem Memory-Pool) wird der Returncode `X'0005' (INVALID ADDRESS OF ACCESS LISTS)` bzw. `X'0007' (INVALID NUMBER OF IOAREA POOL)` ausgegeben und nicht `X'0006' (INVALID ADDRESS OF ACCESS LISTS)` bzw. `X'0008' (INVALID SIZE OF IOAREA POOL)`.

Abhängigkeiten anderer Funktionen

Will der Anwender einen der beim ENABLE ENVIRONMENT bzw. ENABLE IOAREA POOL vom System resident gemachten Speicherbereiche vor dem DISABLE schon freigeben, so wird dies von den entsprechenden Funktionen (RELM, RELMP, DISMP) abgewiesen (Returncode).
Ein RELMP wird auch für Bereiche in Common Memory Pools abgewiesen, die von anderen Tasks, nicht aber von der eigenen Task zu residenten FASTPAM-Bereichen gemacht worden sind.

Will der Anwender eine beim ENABLE ENVIRONMENT angegebene Ereigniskennung vor dem DISABLE ENVIRONMENT schon freigeben, so wird dies ebenfalls von der entsprechenden Funktion DISABLE EVENT ITEM (Makro DISEI) mit Returncode abgewiesen.

Ein USER-CLOSE-ALL wirkt nicht für von FASTPAM geöffnete Dateien.

Layout der Parameterliste

Von einem FPAMSRV Makroauruf wird folgende Parameterliste abgesetzt:

```
FPAMSRV MF=D
1          STACK  PRINT
1          PRINT  NOGEN
2               *,##### PREFIX=F, MACID=PAM #####
1          #INTF REFTYPE=REQUEST,INTNAME=FPAMSRV,INTCOMP=002
1 FPAMPA   DS    0F    BEGIN of PARAMETERAREA           _INOUT
1          FHDR  MF=(C,FPAM),EQUATES=YES
2          DS    0A
2 FPAMFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
2 *
2 FPAMIFID DS    0A              0   INTERFACE IDENTIFIER
2 FPAMFCTU DS    AL2             0   FUNCTION UNIT NUMBER
2 *                                  BIT 15    HEADER FLAG BIT,
2 *                                  MUST BE RESET UNTIL FURTHER NOTICE
2 *                                  BIT 14-12 UNUSED, MUST BE RESET
2 *                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
2 FPAMFCT  DS    AL1             2   FUNCTION NUMBER
2 FPAMFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
2 *
2 FPAMRET  DS    0A              4   GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 FPAMSRET DS    0AL2            4   SUB RETURN CODE
2 FPAMSR2  DS    AL1             4   SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 FPAMR2OK EQU   X'00'                All correct, no additional info
2 FPAMR2NA EQU   X'01'                Successful, no action was necessary
2 FPAMR2WA EQU   X'02'                Warning, particular situation
2 FPAMSR1  DS    AL1             5   SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A    X'00'           FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B    X'01' - X'1F'   PARAMETER SYNTAX ERROR
2 * CLASS C    X'20'           INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D    X'40' - X'7F'   NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E    X'80' - X'82'   WAIT AND RETRY
2 *
2 FPAMRFSP EQU   X'00'                FUNCTION SUCCESSFULLY PROCESSED
2 FPAMRPER EQU   X'01'                PARAMETER SYNTAX ERROR
2 *  3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
```

```
2 FPAMRFNS EQU   X'01'               CALLED FUNCTION NOT SUPPORTED
2 FPAMRFNA EQU   X'02'               CALLED FUNCTION NOT AVAILABLE
2 FPAMRVNA EQU   X'03'               INTERFACE VERSION NOT SUPPORTED
2 *
2 FPAMRAER EQU   X'04'               ALIGNMENT ERROR
2 FPAMRIER EQU   X'20'               INTERNAL ERROR
2 FPAMRCAR EQU   X'40'               CORRECT AND RETRY
2 *  2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 FPAMRECR EQU   X'41'               SUBSYSTEM (SS) MUST BE CREATED
2 *                                  EXPLICITELY BY CREATE-SS
2 FPAMRECN EQU   X'42'               SS MUST BE EXPLICITELY CONNECTED
2 *
2 FPAMRWAR EQU   X'80'               WAIT FOR A SHORT TIME AND RETRY
2 FPAMRWLR EQU   X'81'                    "     LONG       "
2 FPAMRWUR EQU   X'82'               WAIT TIME IS UNCALCULABLY LONG
2 *                                  BUT RETRY IS POSSIBLE
2 *  2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 FPAMRTNA EQU   X'81'               SS TEMPORARILY NOT AVAILABLE
2 FPAMRDH  EQU   X'82'               SS IN DELETE / HOLD
2 *
2 FPAMMRET DS    0AL2           6   MAIN RETURN CODE
2 FPAMMR2  DS    AL1            6   MAIN RETURN CODE 2
2 FPAMMR1  DS    AL1            7   MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XXYYYY')
2 *
2 FPAMRLNK EQU   X'FFFF'             LINKAGE ERROR / REQ. NOT PROCESSED
2 FPAMFHL  EQU   8              8   GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *
1 * SUB RETURN CODE2
1 *
1 FPAMPRIV EQU   X'01'       FASTPAM PRIVILEGE MISSING
1 FPAMRMS  EQU   X'02'       REAL MEMORY SHORTAGE
1 FPAMULE  EQU   X'03'       USER LIMIT EXCEEDED
1 FPAMENR  EQU   X'04'       EXISTING NOT RESIDENT
1 FPAMDSA  EQU   X'05'       DATA SPACE ADDRESS
1 *
1 * MAINCODE
1 *
1 FPAMMFSP EQU   X'0000'     FUNCTION SUCCESSFULLY PROCESSED    = 0
1 FPAMIENN EQU   X'0001'     INVALID ENVIRONMENT NAME           = 1
1 FPAMIIPN EQU   X'0002'     INVALID IOAREA POOL NAME           = 2
1 FPAMIALA EQU   X'0005'     INVALID ADDRESS OF ACCESS LISTS    = 5
1 FPAMIALN EQU   X'0006'     INVALID NUMBER OF ACCESS LISTS     = 6
1 FPAMIIPA EQU   X'0007'     INVALID ADDRESS OF IOAREA POOL     = 7
1 FPAMIIPS EQU   X'0008'     INVALID SIZE OF IOAREA POOL        = 8
1 FPAMISUP EQU   X'0009'     INVALID SHARE UPDATE               = 9
```

```
1 FPAMIMAX EQU   X'000A'     INVALID MAXIMUM IO-LENGTH          = 10
1 FPAMIEVN EQU   X'000B'     INVALID EVENTING                   = 11
1 FPAMIMOD EQU   X'000C'     INVALID MODE                       = 12
1 FPAMIEID EQU   X'000D'     INVALID EVENT-ITEM SHORT-ID        = 13
1 FPAMIBLS EQU   X'000E'     INVALID BLOCKSIZE                  = 14
1 FPAMIENI EQU   X'000F'     INVALID ENVIRONMENT SHORT-ID       = 15
1 FPAMIIPI EQU   X'0010'     INVALID IOAREA POOL SHORT-ID       = 16
1 FPAMIOPI EQU   X'0011'     INVALID OPEN SHORT-ID              = 17
1 FPAMIRES EQU   X'0012'     INVALID RESIDENT                   = 18
1 FPAMILBL EQU   X'0013'     INVALID LAST BLOCK                 = 19
1 FPAMIENV EQU   X'0014'     INVALID ENV-SPECIFICATION          = 20
1 FPAMILRF EQU   X'0015'     INVALID LARGE_FILE-SPECIFICATION   = 21
1 FPAMMIER EQU   X'0028'     INTERNAL ERROR                     = 40
1 FPAMNORE EQU   X'0032'     SPACE NOT RESIDENT                 = 50
1 FPAMDMSE EQU   X'0033'     DMS ERROR DURING OPEN/CLOSE        = 51
1 FPAMRIO  EQU   X'0035'     RUNNING IO                         = 53
1 FPAMOFI  EQU   X'0036'     OPENED FILES                       = 54
1 FPAMSRES EQU   X'0037'     SHORTAGE OF RESOURCES              = 55
1 FPAMTPR  EQU   X'0038'     ENABLE_FROM_TPR_ONLY               = 56
1 FPAMNPRI EQU   X'0039'     NO_PRIVILEGE_FOR_CONNECTION        = 57
1 FPAMDAC@ EQU   X'003B'     DIFFERENT_ACCESS_LISTS_@           = 59
1 FPAMDAC# EQU   X'003C'     DIFFERENT_ACCESS_LISTS_#           = 60
1 FPAMDIP@ EQU   X'003D'     DIFFERENT_IOAREA_POOL_@            = 61
1 FPAMDIPS EQU   X'003E'     DIFFERENT_IOAREA_POOL_SIZE         = 62
1 FPAMDMAX EQU   X'003F'     DIFFERENT_MAXIOLEN                 = 63
1 FPAMDEVE EQU   X'0040'     DIFFERENT_EVENTING                 = 64
1 FPAMDEVI EQU   X'0041'     DIFFERENT_EVENT_ITEM               = 65
1 FPAMDBC  EQU   X'0042'     DIFFERENT_BLKSIZE_IN_CATALOG       = 66
1 FPAMNEIC EQU   X'0046'     NO_EVENT_ITEM_CONNECTION           = 70
1 FPAMEISS EQU   X'0047'     EVENT_ITEM_SCOPE_TO_SMALL          = 71
1 FPAMENEX EQU   X'0048'     ENVIRONMENT_NAME_EXISTING          = 72
1 FPAMINEX EQU   X'0049'     IOAREA_POOL_NAME_EXISTING          = 73
1 FPAMODS  EQU   X'004A'     OVERLAPPING_DIV_SPACE              = 74
1 FPAMOFS  EQU   X'004B'     OVERLAPPING_FASTPAM_SPACE          = 75
1 FPAMALNA EQU   X'004C'     ACCESS_LISTS_NOT_ALLOCATED         = 76
1 FPAMNPAF EQU   X'004D'     NO_PAM_FILE                        = 77
1 FPAMLBIN EQU   X'004E'     LAST BLOCK BUT INPUT               = 78
1 FPAMRFA  EQU   X'0050'     RFA_NOT_SUPPORTED                  = 80
1 FPAMSPD  EQU   X'0051'     SPD_NOT_SUPPORTED                  = 81
1 FPAMPPD  EQU   X'0052'     PPD_NOT_SUPPORTED                  = 82
1 FPAMTAPE EQU   X'0053'     TAPE_NOT_SUPPORTED                 = 83
1 FPAMDUMM EQU   X'0054'     DUMMY_FILE_NOT_SUPPORTED           = 84
1 FPAMSECA EQU   X'0055'     SEC_ALLOCATION_TOO_SMALL           = 85
1 FPAMTPRE EQU   X'0056'     TPR_ENVIRONMENT_OR_IOAREA_POOL     = 86
1 FPAMIPAL EQU   X'0057'     IOAREA_POOL_NOT_ALLOCATED          = 87
1 FPAMBLKN EQU   X'0058'     BLKCTRL_NOT_SUPPORTED              = 88
1 FPAMTPRO EQU   X'0059'     TPR_OPEN_ID                        = 89
1 FPAMWRCN EQU   X'005A'     WRCHK_NOT_SUPPORTED                = 90
```

```
1 FPAMMPNF EQU   X'005B'     MEMORY_POOL_NOT_FIXED              = 91
1 *
1 * &P.PAM FUNCTIONS:
1 *
1 FPAMENEV EQU   1           ENABLE ENVIRONMENT
1 FPAMDIEV EQU   2           DISABLE ENVIRONMENT
1 FPAMENIP EQU   3           ENABLE IOAREA POOL
1 FPAMDIIP EQU   4           DISABLE IOAREA POOL
1 FPAMOPEN EQU   5           OPEN FILE
1 FPAMCLOS EQU   6           CLOSE FILE
1 *
1 * INPUT/OUTPUT PARAMETER
1 *
1 FPAMENID DS    F           ENVIRONMENT SHORT-ID
1 FPAMIPID DS    F           IOAREA POOL SHORT-ID
1 FPAMOPID DS    F           OPEN SHORT-ID
1 *
1 * OUTPUT PARAMETER
1 *
1 FPAMDMSC DS    XL4         DMS-CODE
1 *
1 * INPUT PARAMETER
1 *
1 FPAMENNA DS    CL54        ENVIRONMENT NAME
1 FPAMIPNA DS    CL54        IOAREA POOL NAME
1 FPAMLINK DS    CL8         LINK
1 FPAMFILE DS    CL54        FILENAME
1 FPAMLARF DS    AL1         LARGE_FILE
1 FPAMFRBD EQU   0           LARGE_FILE=FORBIDDEN
1 FPAMALWD EQU   1           LARGE_FILE=ALLOWED
1 FPAMENV  DS    AL1         ENV
1 FPAMHOST EQU   1           ENV=HOST
1 FPAMXCS  EQU   2           ENV=XCS
1 FPAMACLA DS    A           ADDRESS OF ACCESS LISTS
1 FPAMACLN DS    F           NUMBER OF ACCESS LISTS
1 FPAMOFF  DS    A           ADDR. OF IOAREA POOL WITHIN ADDRESS SPACE
1 FPAMALET DS    F           ALET OF DATA SPACE
1 FPAMIPS  DS    F           SIZE OF IOAREA POOL (4K)
1 FPAMEID  DS    F           EVENT-ITEM SHORT-ID
1 FPAMLABL DS    F           LAST BLOCK NUMBER
1 FPAMMODE DS    AL1         OPEN MODE
1 FPAMINPT EQU   1           MODE=INPUT
1 FPAMINOT EQU   2           MODE=INOUT
1 FPAMOUTI EQU   3           MODE=OUTIN
1 FPAMSUPD DS    AL1         SHARE UPDATE
1 FPAMSUNO EQU   1           SHARUPD=NO
1 FPAMSUYE EQU   2           SHARUPD=YES
1 FPAMMAXL DS    AL1         MAXIMUM IO-LENGTH
```

```
1 FPAMMINS EQU   0           IO-LENGTH NOT SPECIFIED
1 FPAMMINI EQU   1           IO-LENGTH=4K
1 FPAMMAXI EQU   8           IO-LENGTH=32K
1 FPAMEVEN DS    AL1         EVENTING
1 FPAMEVNS EQU   0           EVENTING=NOT_SPECIFIED
1 FPAMEVNO EQU   1           EVENTING=NO
1 FPAMEVYE EQU   2           EVENTING=YES
1 FPAMRESI DS    AL1         RESIDENT
1 FPAMRENS EQU   0           RESIDENT=NOT_SPECIFIED
1 FPAMRENO EQU   1           RESIDENT=NO
1 FPAMREYE EQU   2           RESIDENT=YES
1 FPAMBLS  DS    FL1         BLOCKSIZE
1 FPAM#    EQU   *-FPAMPA LENGTH of PARAMETERAREA
```

FSTAT – Kataloginformation anfordern

Makrotyp: S-Typ (E-Form, L-Form/C-Form/D-Form) (siehe Seite 870)

Der FSTAT-Makroaufruf informiert über Katalogeinträge. Der Benutzer kann Informationen über einzelne oder mehrere Dateien, Dateigenerationen, Dateigenerationsgruppen oder alle Dateien einer Benutzerkennung anfordern.

Der Benutzer kann sich über alle Dateien unter seiner Benutzerkennung informieren, sowie über alle Dateien anderer Benutzer, auf die er zugreifen darf (vgl. die Selektionsoperanden SHARE, BASACL, OWNERAR, GROUPAR, OTHERAR, ACL, GUARDS und PROTACT).

Die Auswahl der Dateien, über die der Benutzer Informationen wünscht, erfolgt über:

– den Pfadnamen. Auswahlkriterien sind Katalogkennung, Benutzerkennung und Dateiname (voll- oder teilqualifiziert, mit oder ohne Wildcards).
  Ohne Angabe eines Pfadnamens werden alle permanenten Dateien der eigenen Benutzerkennung aus dem Standardkatalog des lokalen Rechners ausgewählt.
  Temporäre Dateien müssen mit dem Tempfile-Präfix (# oder @) angesprochen werden.
– Die über „pfadname“ getroffene Dateiauswahl kann der Benutzer über die Selektionsoperanden weiter einschränken. Es werden nur Dateien ausgewählt, die die in den Selektionsoperanden beschriebenen Dateimerkmale aufweisen. Wird ein Selektionsoperand nicht angegeben bzw. der Wert ANY (soweit möglich) spezifiziert, erfolgt die Auswahl unabhängig von dem entsprechenden Dateimerkmal.

Umfang und Aufbau der Informationen, die für die ausgewählten Dateien in den Ausgabebereich zu übertragen sind, kann der Benutzer im Operanden OUTPUT bestimmen:

– nur Rückinformation über die Makroausführung (keine Information im Ausgabebereich)
– nur die Namen der ausgewählten Dateien
– statistische Informationen (z.B. Anzahl der ausgewählten Dateien je Datenträgerart)
– Kataloginformationen je ausgewählter Datei

Bei Ausgabe der Kataloginformationen kann der Benutzer die Ausgabe auf einen oder mehrere Informationsblöcke, in denen Dateimerkmale zu logischen Einheiten zusammengefasst sind, beschränken (Operand CEINFO):

– Voreinstellung: alle Informationsblöcke (ohne FTAM-Information)
– History-Block
– Security-Block
– Status-Block
– Backup-Block
– Organization-Block
– Allocation-Block
– Volumes-Block
– Volume-Extents-Block
– Index-Info-Block
– FTAM-Information

Als Informationsquelle kann der Benutzer auswählen (siehe Operand FROM):

– den Standardkatalog der Benutzerkennung
– alle Kataloge der lokalen Pubsets
– VTOC einer Privatplatte oder eines Net-Storage-Volumes

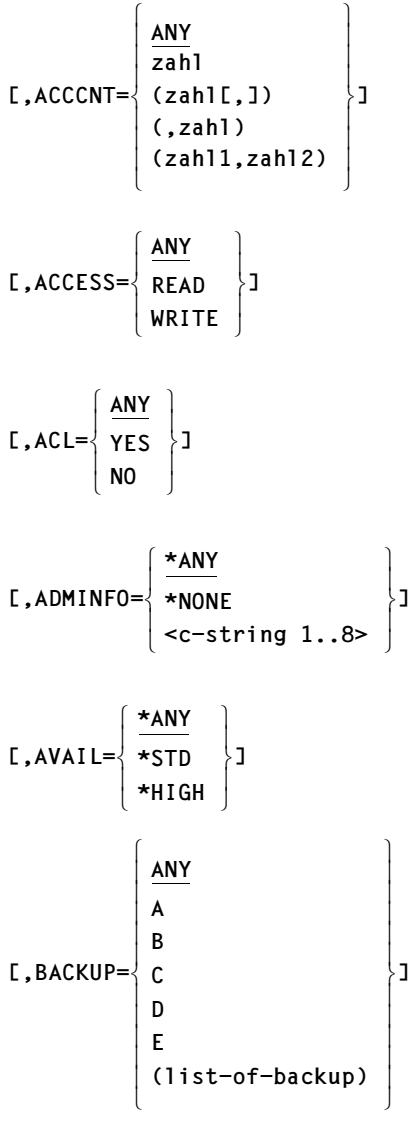
Der FSTAT-Makroaufruf wird in fünf Versionen unterstützt (siehe auch Operand VERSION):

– Version 4		für BS2000-Versionen ≥ BS2000/OSD-BC V9.0
– Version 3		für BS2000-Versionen ≥ BS2000/OSD-BC V3.0
– Version 2		für BS2000-Versionen ≥ BS2000/OSD-BC V1.0
– Version 1	≙ Version 800	für BS2000-Versionen ≥ V8.0
– Version 0	≙ Version 710	für BS2000-Versionen < V8.0

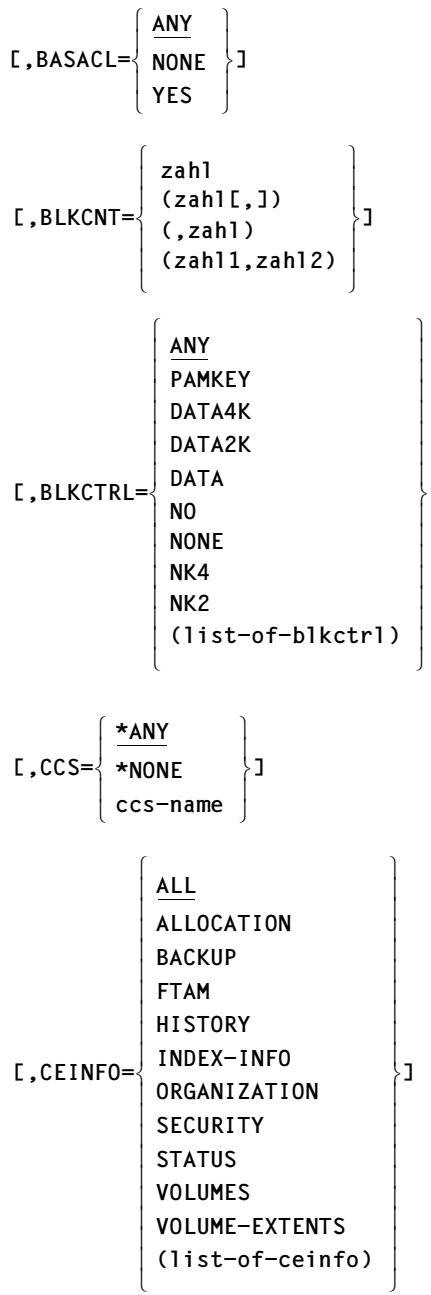
Voreinstellung: Version 0

Die vorliegende Beschreibung des FSTAT-Makroaufrufs bezieht sich auf den Funktionsumfang der Version 4, die im Makroaufruf explizit mit VERSION=4 vereinbart werden muss. In der Tabelle „Versionsunterschiede – VERSION=0/1/2/3/4“ auf Seite 666 sind für die Versionen 0 bis 3 alle Abweichungen zur beschriebenen Version zusammengefasst.

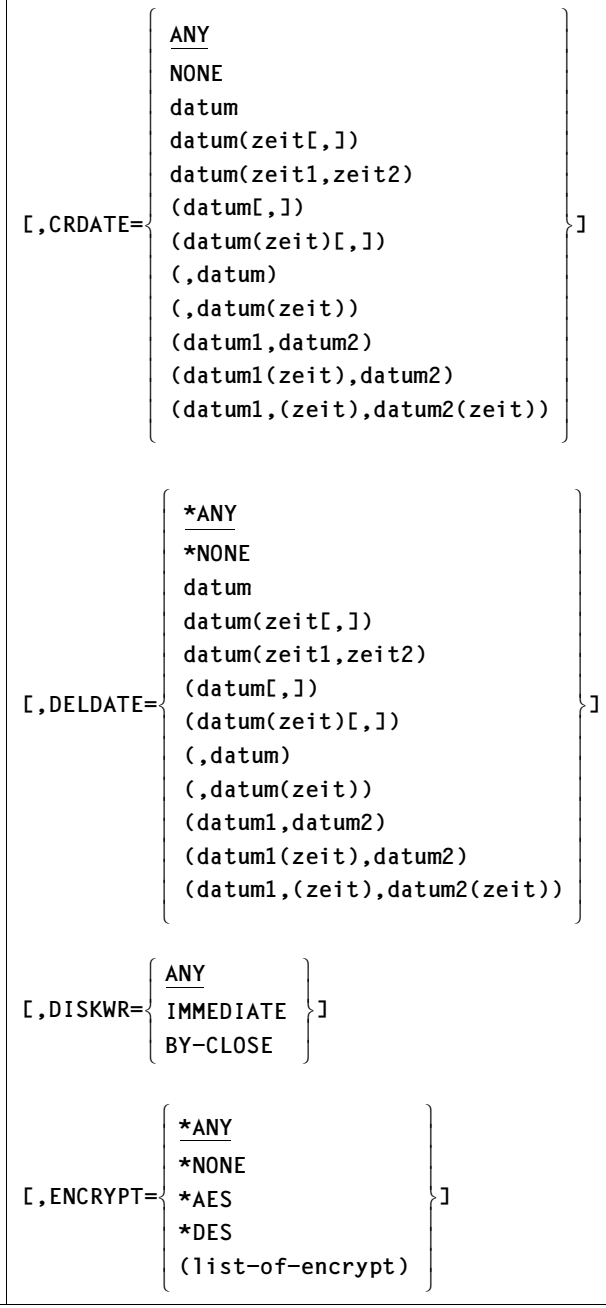
Format

Operation	Operanden
FSTAT	[pfadname] 

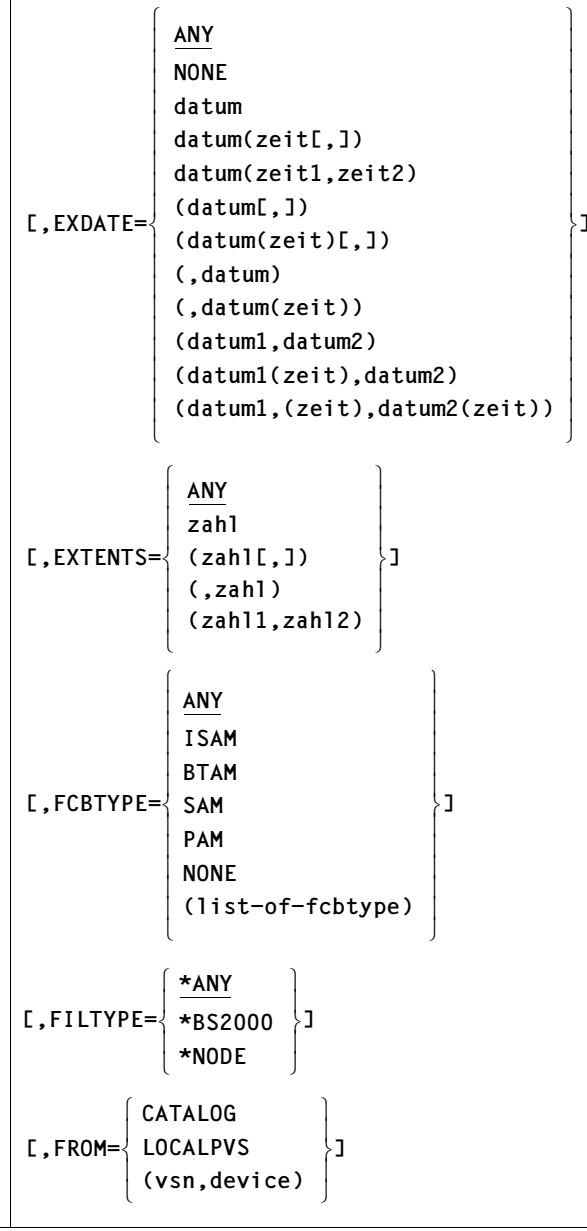
(Teil 1 von 11)

Operation	Operanden
	

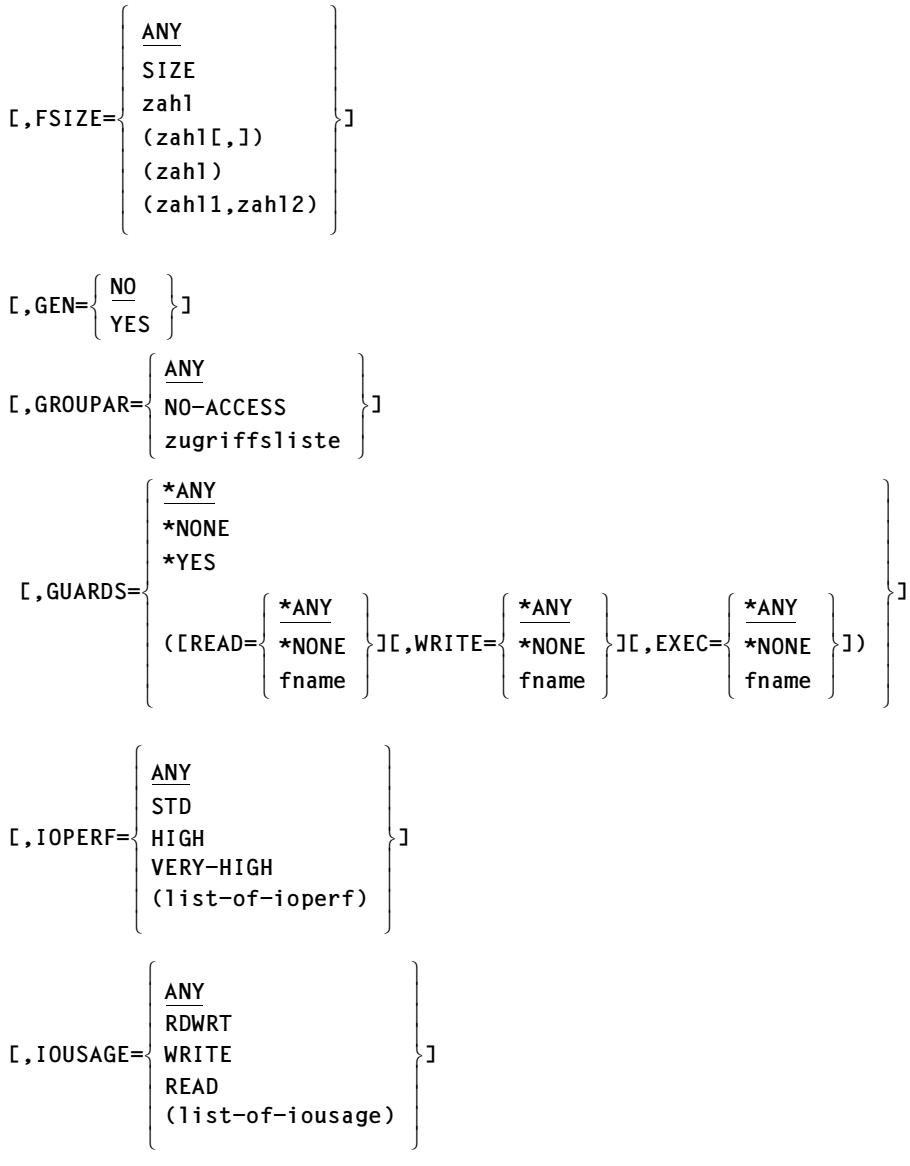
(Teil 2 von 11)

Operation	Operanden
	

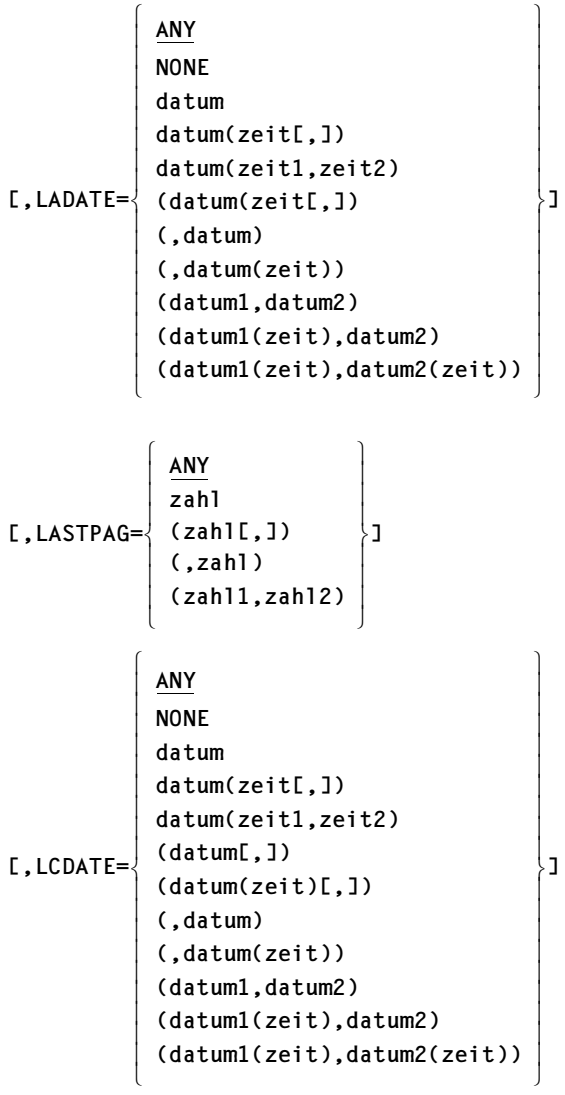
(Teil 3 von 11)

Operation	Operanden
	

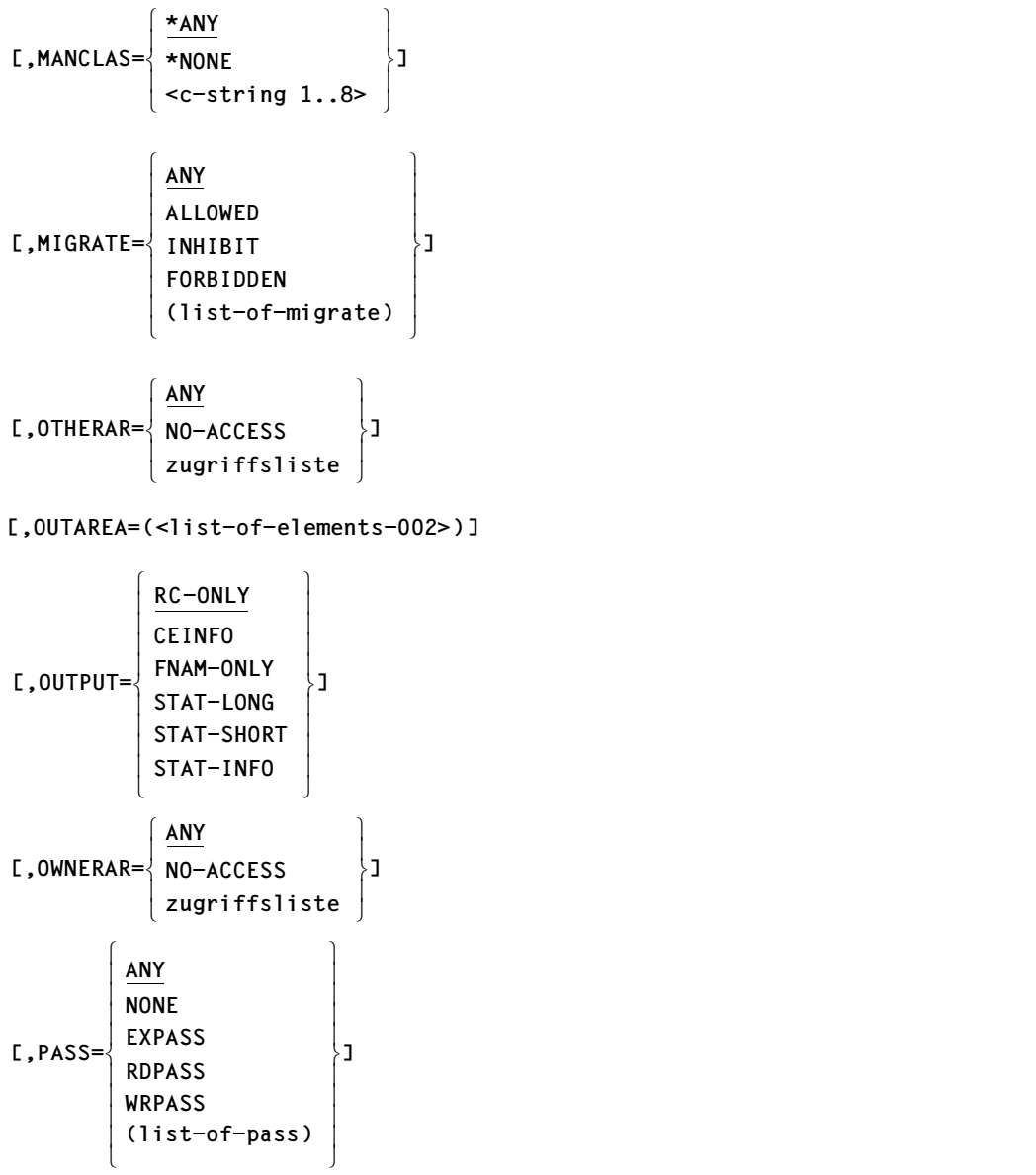
(Teil 4 von 11)

Operation	Operanden
	

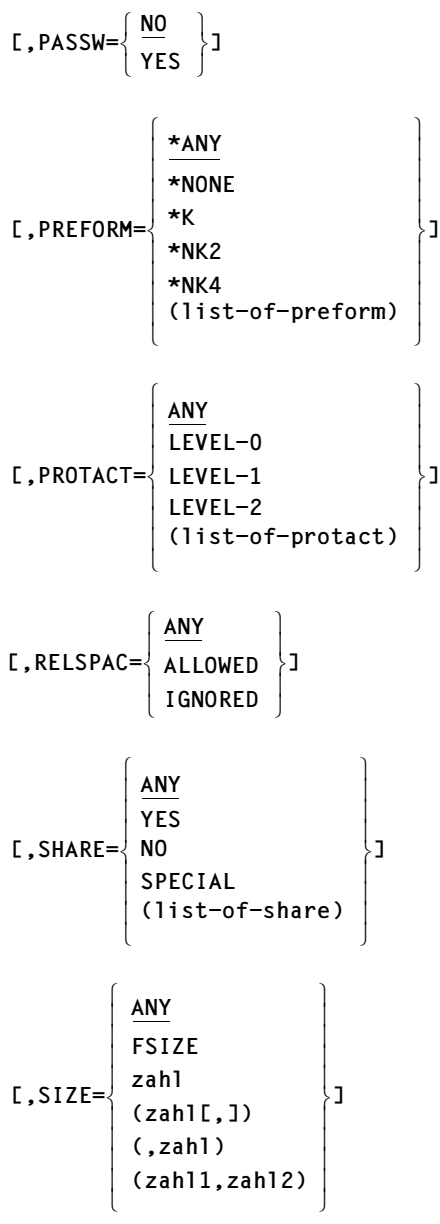
(Teil 5 von 11)

Operation	Operanden
	

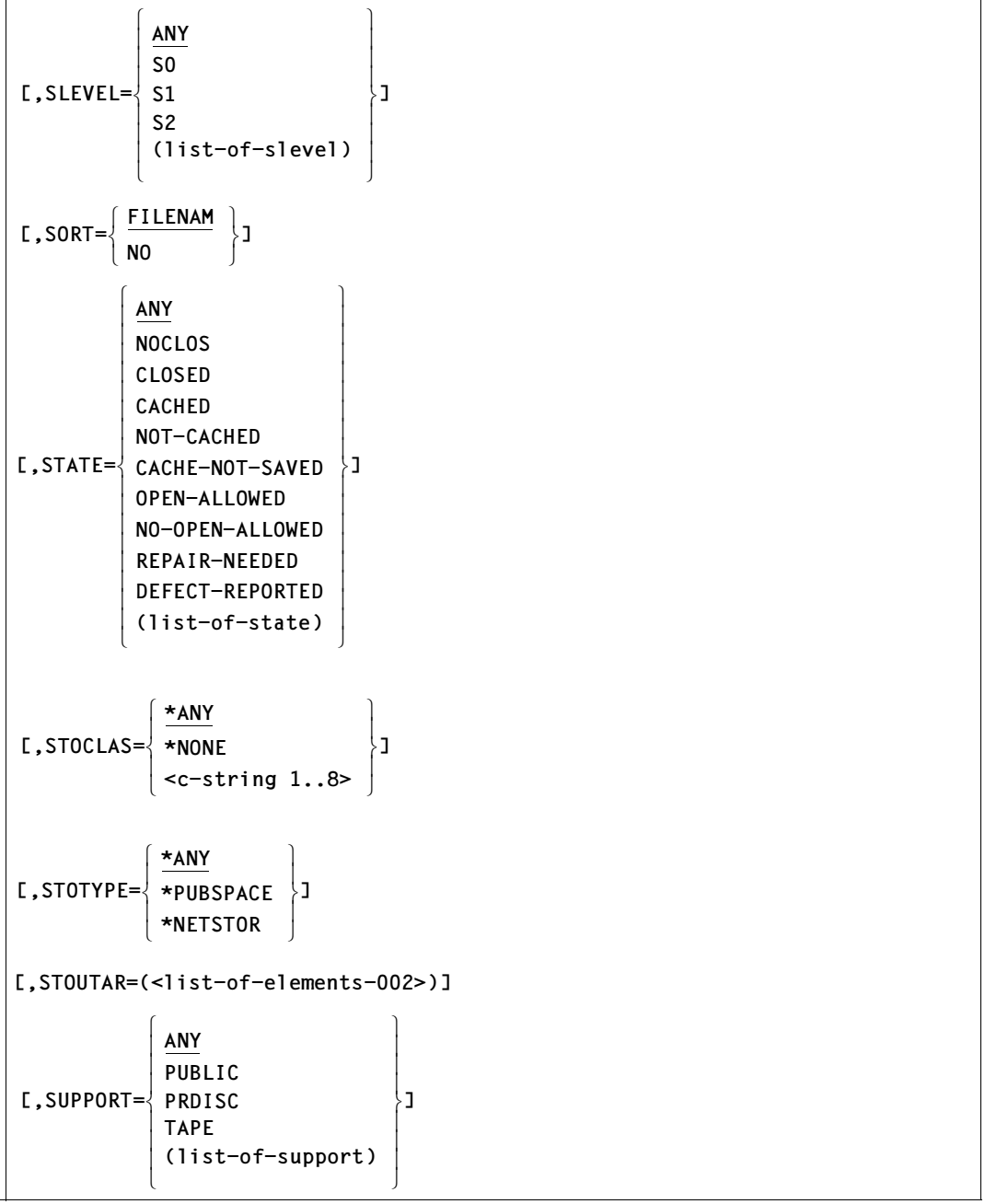
(Teil 6 von 11)

Operation	Operanden
	

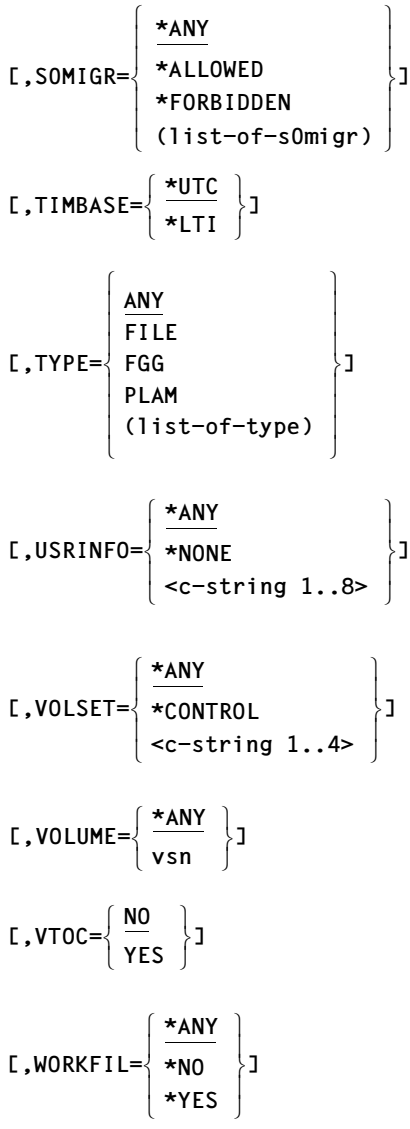
(Teil 7 von 11)

Operation	Operanden
	

(Teil 8 von 11)

Operation	Operanden
	

(Teil 9 von 11)

Operation	Operanden
	

(Teil 10 von 11)

Operation	Operanden
	[,WTQUIET={ *YES / *NO }]
	[,XPAND={ PLSHORT / PLLONG / OUTPUT / (PLSHORT,OUTPUT) / (PLLONG,OUTPUT) }]
	[,MF=L],VERSION={ 0 / 1 / 2 / 3 / 4 }[,PREFIX=pre]
	MF=(E,{ adr / (r) }),VERSION={ 0 / 1 / 2 / 3 / 4 }
	[,MF={ C / D },[VERSION={ 0 / 1 / 2 / 3 / 4 }][,PREFIX=pre][,XPAND={ PLSHORT / PLLONG / OUTPUT / (PLSHORT,OUTPUT) / (PLLONG,OUTPUT) }]

(Teil 11 von 11)

Operandenbeschreibung

pfadname
bezeichnet den Pfadnamen der Datei(en), über die informiert werden soll, mit:
<filename 1..54 with-wild(80) without-gen>

Temporäre Dateien werden nicht berücksichtigt.

Pfadname bedeutet [:catid:][$userid][dateiname]

catid
Katalogkennung;
Default-Catid: die der Benutzerkennung zugeordnete Katalogkennung

userid
Benutzerkennung;
„$userid." bezeichnet alle Dateien dieses Benutzers. Wird die Benutzerkennung eines anderen Benutzers angegeben, werden nur Informationen über Dateien ausgegeben, auf die der Aufrufer des Makros zugreifen darf.
Default-Userid: eigene Benutzerkennung, d.h. die des SET-LOGON-PARAMETERS- bzw. LOGON-Kommandos

dateiname
Voll- oder teilqualifizierter Dateiname von permanenten oder temporären Dateien, von Dateigenerationen oder Dateigenerationsgruppen.

Wildcard-Angabe (Musterzeichen)
Der nichtprivilegierte Benutzer darf Musterzeichen nur in der Catid und im Dateinamen verwenden; der Systemverwalter auch in der Benutzerkennung (zu Musterzeichen siehe Seite 867).
Platzhalter können nicht die Begrenzer der Namensteile cat (Doppelpunkte) und user ($-Zeichen und Punkt) ersetzen.

ACCCNT
Informiert über alle Dateien, auf die so oft, wie angegeben, zugegriffen wurde.
Der Zugriffszähler kann Werte von 0 bis 2147483647 annehmen.

= ANY
Der Zugriffszähler ist kein Auswahlkriterium.

= zahl
Informiert über Dateien, deren Zugriffszähler genau den angegebenen Wert besitzt.

= (zahl[,])
Informiert über Dateien, deren Zugriffszähler größer oder gleich dem angegebenen Wert ist.

= (,zahl)
Informiert über Dateien, deren Zugriffszähler kleiner oder gleich dem angegebenen Wert ist.

= (zahl1,zahl2)
Informiert über Dateien, deren Zugriffszähler in dem angegebenen Intervall liegt (zahl1 ≤ Zugriffszähler ≤ zahl2).

ACCESS
Informiert abhängig von der Zugriffsart über Dateien/Dateigenerationen.

= ANY
Die Zugriffsart ist kein Auswahlkriterium.

= READ
Informiert über Dateien/Dateigenerationen, für die nur Lesezugriff gestattet ist.

= WRITE
Informiert über Dateien/Dateigenerationen, für die auch Schreibzugriff gestattet ist.

ACL
Informiert über Dateien in Abhängigkeit davon, ob sie durch einen ACL-Eintrag geschützt werden.

= ANY
Der ACL-Eintrag ist kein Auswahlkriterium.

= NO
Informiert über alle Dateien, die nicht durch einen ACL-Eintrag geschützt sind.

= YES

Seit SECOS V4.0 wird die Zugriffskontrolle über ACL nicht mehr unterstützt. Der ACL-Eintrag enthält deshalb im Normalfall den Wert NO (kein ACL-Schutz).

ADMINFO
Informiert abhängig von der Systemverwalter-Metainformation über Dateien/Dateigenerationen.

= *ANY
Die Systemverwalter-Metainformation ist kein Auswahlkriterium.

= *NONE
Informiert über Dateien, die keine Systemverwalter-Metainformation besitzen.

= <c-string 1..8>
Informiert über Dateien mit der angegebenen Systemverwalter-Metainformation.

AVAIL
Informiert abhängig von der Ausfallsicherheit über Dateien/Dateigenerationen.

= *ANY
Die Ausfallsicherheit ist kein Auswahlkriterium.

= *STD
Informiert über Dateien, die sich nicht auf einem Volume-Set oder einem SF-Pubset mit hoher Ausfallsicherheit befinden.

= *HIGH
Informiert über Dateien, die sich auf Platten mit hoher Ausfallsicherheit befinden (DRV-Pubset).

BACKUP
Informiert über Dateien/Dateigenerationsgruppen, für die die angegebene ARCHIVE- bzw. HSMS-Sicherungsstufe festgelegt wurde.

= ANY
Die Sicherungsstufe ist kein Auswahlkriterium.

= A
Informiert über Dateien/FGG mit dem Merkmal BACKUP=A.

= B
Informiert über Dateien/FGG mit dem Merkmal BACKUP=B.

= C
Informiert über Dateien/FGG mit dem Merkmal BACKUP=C.

= D
Informiert über Dateien/FGG mit dem Merkmal BACKUP=D.

= E
Informiert über Dateien/FGG mit dem Merkmal BACKUP=E.

= (list-of-backup)
In einer Liste können auch mehrere Sicherungstufen angegeben werden. Dann werden alle Dateien/FGG berücksichtigt, die einer dieser Bedingungen genügen (Oder-Verknüpfung).

BASACL
Informiert über Dateien in Abhängigkeit von einer vereinbarten BASIC-ACL.

= ANY
Die BASIC-ACL ist kein Auswahlkriterium.

= NONE
Informiert über alle Dateien, für die kein BASIC-ACL-Eintrag definiert ist.

= YES
Informiert über alle Dateien, für die ein BASIC-ACL-Eintrag definiert ist.

BLKCNT
Nur für Banddateien
Informiert über Dateien in Abhängigkeit von der Anzahl der Blöcke auf Band.

= ANY
Die Anzahl der Blöcke auf Band ist kein Auswahlkriterium.

= zahl
Informiert über alle Banddateien mit genau der angegebenen Anzahl von Blöcken.

= (zahl[,])
Informiert über alle Banddateien, deren Anzahl von Blöcken größer oder gleich dem angegebenen Wert ist.

= (,zahl)
Informiert über alle Banddateien, deren Anzahl von Blöcken kleiner oder gleich dem angegebenen Wert ist.

= (zahl1,zahl2)
Informiert über alle Banddateien, deren Anzahl von Blöcken in dem angegebenen Intervall liegt.
Als Wert sind ganze Zahlen aus dem Intervall $0 \leq$ wert ≤ 2147483647 erlaubt.

BLKCTRL
Informiert über Dateien, abhängig von dem Blockformat, mit dem die Datei gespeichert wurde (im FILE- oder FCB- Makroaufruf mit dem Operanden BLKCTRL definiert). Das Blockformat wird beim Erstellen der Datei festgelegt und bezieht sich auf die Existenz und Position des Blockkontrollfeldes, das die Verwaltungsinformationen für die PAM-Seite enthält.

= ANY
Das Dateiformat ist kein Auswahlkriterium.

= DATA
Informiert über alle Dateien, bei denen das Blockkontrollfeld am Anfang des Datenblocks steht. Die Dateien wurden mit BLKCTRL=DATA erstellt (siehe FILE-Makro).

= PAMKEY
Informiert über alle Dateien, die für das Blockkontrollfeld einen separaten PAM-Schlüssel nutzen, d.h., die Blockkontroll-Information steht in einem separaten Schlüsselfeld außerhalb des PAM-Blockes. Die Dateien wurden mit BLKCTRL=PAMKEY erstellt (siehe FILE-Makro).

= NO
Informiert über alle Dateien, die kein Blockkontrollfeld enthalten. Die Dateien wurden mit BLKCTRL=NO erstellt (siehe FILE-Makro).

= NONE
Informiert über alle Dateien, für die kein BLKCTRL-Wert definiert wurde, d.h. Dateien, die noch nicht eröffnet wurden.

= DATA2K
Informiert über alle Dateien, die mit BLKCTRL=DATA2K erstellt wurden (siehe FILE-Makro).

= DATA4K
Informiert über alle Dateien, die mit BLKCTRL=DATA4K erstellt wurden (siehe FILE-Makro).

= NK2
Informiert über alle NK2-Dateien (können auf NK2-Datenträgern abgelegt werden).

= NK4
Informiert über alle NK4-Dateien (können auf NK4-Datenträgern abgelegt werden).

= (list-of-blkctrl)
Informiert über alle Dateien, die einem der angegebenen Blockformate entsprechen. Innerhalb der Liste können alle Werte außer ANY angegeben werden.

CCS
Informiert über alle Dateien entsprechend der angegebenen Codiertabelle (Coded Character Set).
In der Codiertabelle ist festgelegt, wie die Zeichen eines nationalen Zeichensatzes binär abzuspeichern sind. Der festgelegte Zeichensatz beeinflusst z.B. die Bildschirmdarstellung von Zeichen, Sortierreihenfolge usw. (siehe Handbuch „XHCS“ [22])

= *ANY
Die Kodiertabelle ist kein Auswahlkriterium.

= *NONE
Nur Dateien, für die kein Zeichensatz definiert ist, werden ausgewählt.

= ccs-name
Nur Dateien mit dem angegebenen Zeichensatz werden ausgewählt.

CEINFO
Die Informationen aus dem Katalog sind in Informationsblöcken logisch zusammengefasst. Ausgegeben werden die Informationsblöcke, die der Benutzer explizit auswählt.

Mithilfe des Operanden CEINFO kann man die Generierung der DSECT bzw. CSECT steueren. Wird CEINFO nicht angegeben oder wird CEINFO=ALL gesetzt, so werden alle Ausgabeblöcke generiert (bis auf den FTAM-Block). Ansonsten werden diejenigen Ausgabeblöcke generiert, die bei CEINFO spezifiziert sind.

= ALL
Überträgt für die ausgewählten Dateien sämtliche im Katalog gespeicherten Informationen in den Ausgabebereich (OUTAREA).
Die Informationen sind folgenden Informationsblöcken zugeordnet: HISTORY / SECURITY / STATUS / BACKUP / ORGANIZATION / ALLOCATION / VOLUMES / VOLUME-EXTENTS (bzw. INDEX-INFO)

Die Ausgabe der File-Transfer-Information FTAM kann über CEINFO=ALL nicht erreicht werden. Diese nur für den File-Transfer relevante Information wird nur mit CEINFO=FTAM ausgegeben.

= HISTORY
Überträgt für die ausgewählten Dateien den History-Block in den Ausgabebereich, d.h. alle Dateimerkmale, die „historischen“ Charakter haben:

- Zugriffszähler
- Datum des letzten Zugriffs
- Zeitpunkt des letzten Zugriffs
- Datum des letzten schreibenden Zugriffs
- Tageszeit des letzten schreibenden Zugriffs
- Erstellungsdatum
- Tageszeit der Erstellung
- Anzahl der Speicherplatzerweiterungen

} Basis siehe Operand TIMBASE

= SECURITY
Überträgt für die ausgewählten Dateien den Security-Block in den Ausgabebereich, d.h. alle Dateimerkmale, die den Dateischutz betreffen (die Bytes, welche für die Dateischutzwörter bestimmt sind, werden auf binär 0 gesetzt. Ausnahme siehe Operand „PASSW“):

- Art des Zugriffs (Standard-Zugriffskontrolle)
- Angabe zum Dateischutz mit ACL (nur aus Kompatibilitätsgründen vorhanden)
- Dateiüberwachung
- automatische Datenzerstörung beim Löschen
- Schutz mit Ausführungskennwort
- Datum, an dem die Datei wieder verändert werden darf
- Zeitpunkt bezogen auf EXPIR-DATE
- Schutz mit Lese-GUARD

– Schutz mit Schreib-GUARD
– Schutz mit Ausführungs-GUARD
– Zugriffsrechte der Benutzerklasse „Group“ (BASIC-ACL)
– Zugriffsrechte der Benutzerklasse „Others“ (BASIC-ACL)
– Zugriffsrechte des Dateieigentümers (BASIC-ACL)
– Schutz mit Lesekennwort
– Schutz gegen Speicherplatzfreigabe
– Angabe zur Mehrbenutzbarkeit (Standard-Zugriffskontrolle)
– Schutz mit Schreibkennwort

= STATUS
Überträgt für die ausgewählten Dateien den Status-Block in den Ausgabebereich, d.h. alle Dateimerkmale, die besondere Dateieigenschaften betreffen. Dies sind:

– SPECIAL-ACCOUNTING-BIT
– OPEN-CLOSE-Indikator
– REPAIR-Indikator
– PSEUDO-CLOSE-Indikator
– VERIFY-IS-FORBIDDEN-Indikator
– LOCKS (RELEASE-LOCK; ERASE-LOCK; OUTPUT-LOCK; CATALOG-LOCK; SPD-LOCK)

= BACKUP
Überträgt für die ausgewählten Dateien den Backup-Block in den Ausgabebereich, d.h. alle Dateimerkmale, die die Dateisicherung betreffen:

– Backup-Stufe für Archive bzw. HSMS
– Angabe, ob die Datei migriert werden darf
– Angabe, ob die Datei immer vollständig zu sichern ist
– Version als internes ARCHIVE-Merkmal

= ORGANIZATION
Überträgt für die ausgewählten Dateien den Organization-Block in den Ausgabebereich, d.h. alle Dateimerkmale, die die Dateiorganisation betreffen:

– Blockzähler (Banddateien)
– Pufferverschiebung (Banddateien)
– Blocktyp (Standard- oder Nichtstandardblock)
– Codiertabelle (CCS) bei XHCS-Unterstützung
– Codeangabe für Banddateien
– Eignung der Datei zur Bearbeitung in einem Cache
– Dateifolgenummer (Banddatei)
– Zugriffsmethode bei Erstellung der Datei
– Performance-Anforderung bei Dateibearbeitung
– Art der Ein-/Ausgabeoperationen für die Performance-Anforderung
– Länge des ISAM-Schlüssels
– Position des ISAM-Schlüssels

– Standardversion der Kennsätze (Banddatei)
– Länge der logischen ISAM-Markierung
– Weitergabe der ISAM-Wertmarkierung
– Satzformat
– Satzlänge
– Länge der ISAM-Wertmarkierung
– Dateityp auf Net-Storage
– Dateigröße von Node-Files

= ALLOCATION
Überträgt für die ausgewählten Dateien den Allocation-Block in den Ausgabebereich, d.h. alle Dateimerkmale, die die Speicherplatzbelegung betreffen:

– Gerätetyp für den Datenträger
– Gesamtanzahl der Extents für die Datei
– höchste belegte PAM-Seite (Last Page Pointer)
– Sekundärzuweisung für Dateierweiterung
– Speicherebene für migrierte Dateien
– Datenträgertyp
– Archivnummer des belegten Datenträgers
– letztes gültiges Byte auf der letzten logischen Seite der Datei (Last Byte Pointer: nur für PAM-Datei oder Node-File)

= VOLUMES
Überträgt für die ausgewählten Dateien die Volume-Tabellen in den Ausgabebereich. Diese enthalten folgende Informationen:

– VOLUME
– DEVICE
– #EXTENTS
– Indikator, ob es sich um ein ISAM-INDEX- oder ein ISAM-DATEN-Volume handelt

= VOLUME-EXTENTS
Überträgt für die ausgewählten Dateien die Volume-Tabellen und die Extent-Listen der Datei in den Ausgabebereich.

= INDEX-INFO
Überträgt für die ausgewählten Dateien den INDEX-INFO-Block in den Ausgabebereich, d.h. alle Dateimerkmale, die die Dateigenerationsgruppe betreffen:

– Basiswert für relative Generationsnummern
– jüngste bzw. zuletzt katalogisierte Dateigeneration
– ältestete existierende Dateigeneration
– Maximalzahl der gleichzeitig katalogisierten Generationen
– Vorgehensweise bei Erreichen der Maximalzahl

= FTAM
Überträgt für die ausgewählten Dateien den FTAM-Block in den Ausgabebereich, d.h. alle Dateimerkmale, die den File-Transfer betreffen.
Dieser Ausgabeblock ist nur für File-Transfer relevant, weshalb der FT-Block bei CEINFO=ALL nicht ausgegeben wird.

= (list-of-ceinfo)
In einer Liste können alle Operandenwerte außer FTAM und ALL angegeben werden. Die entsprechenden Informationblöcke werden in den Ausgabebereich übertragen. Die Reihenfolge der Angaben spielt keine Rolle.

CRDATE
Der Anwender kann über das Erstellungsdatum die Dateien auswählen, die bearbeitet werden sollen.

Datumsangaben kann der Anwender durch eine Zeitangabe ergänzen. Die Regeln für die Datums- und Zeitangaben sind auf Seite 868 beschrieben.
Bereichsangaben gelten jeweils einschließlich der angegebenen Grenzen.

= ANY
Das Erstellungsdatum ist kein Auswahlkriterium.

= NONE
Informiert über alle Dateien, für die noch kein Erstellungsdatum im Katalog eingetragen ist, d.h. Dateien, die noch nicht eröffnet wurden.

= datum
Informiert über alle Dateien, die an dem angegebenen Tag erstellt wurden.

= (datum[,])
Informiert über alle Dateien, die seit dem angegebenen Datum erstellt wurden (Erstellungsdatum ≥ datum).

= (,datum)
Informiert über alle Dateien, die bis zu dem angegebenen Datum erstellt wurden (Erstellungsdatum ≤ datum).

= (datum1,datum2)
Informiert über alle Dateien, die während des angegebenen Zeitraums erstellt wurden (datum1 ≤ Erstellungsdatum ≤ datum2).

= datum(zeit[,])
Informiert über alle Dateien, die an dem angegebenen Tag und ab der angegebenen Uhrzeit erstellt wurden.

= datum(zeit1,zeit2)
Informiert über alle Dateien, die an dem angegebenen Tag und innerhalb des angegebenen Zeitraums erstellt wurden.

= (datum(zeit)[,])
Informiert über alle Dateien, die ab dem angegebenen Tag und ab der angegebenen Uhrzeit erstellt wurden.

= (,datum(zeit))
Informiert über alle Dateien, die vor dem angegebenen Tag und ab der angegebenen Uhrzeit erstellt wurden.

= (datum1(zeit),datum2(zeit))
Informiert über alle Dateien, die in dem angegebenen Zeitraum erstellt wurden. Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.

DELDATE
Der Anwender kann über das DELETION-DATE (Zeitpunkt, ab dem die Datei ohne Berücksichtigung der Schutzattribute gelöscht werden darf) die Dateien auswählen, die bearbeitet werden sollen.

Datumsangaben kann der Anwender durch eine Zeitangabe ergänzen. Hierbei ist zu beachten, dass derzeit als Löschzeitpunkt immer die Uhrzeit 00:00:00 im Dateikatalog eingetragen ist.
Die Regeln für die Datums- und Zeitangaben sind auf Seite 868 beschrieben. Bereichsangaben gelten jeweils einschließlich der angegebenen Grenzen.

= *ANY
Das DELETION-DATE ist kein Auswahlkriterium.

= *NONE
Informiert über alle Dateien, für die noch kein DELETION-DATE im Katalog eingetragen ist.

= datum
Informiert über alle Dateien, für die das angegebene DELETION-DATE vereinbart ist.

= (datum[,])
Informiert über alle Dateien, deren DELETION-DATE größer oder gleich dem angegebenen Datum ist.

= (,datum)
Informiert über alle Dateien, deren DELETION-DATE kleiner oder gleich dem angegebenen Datum ist.

= (datum1,datum2)
Informiert über alle Dateien, deren DELETION-DATE innerhalb des angegebenen Zeitraums liegt (datum1 ≤ DELETION-DATE ≤ datum2).

= datum(zeit[,])
Informiert über alle Dateien, für die das angegebene DELETION-DATE vereinbart ist und die Uhrzeit des Löschens größer oder gleich der angegebenen Zeit ist. Die Löschzeit (Uhrzeit bezogen auf das Löschdatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= datum(zeit1,zeit2)
Informiert über alle Dateien, für die das angegebene DELETION-DATE vereinbart ist und die Uhrzeit der Freigabe innerhalb des angegebenen Zeitintervall liegt. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (datum(zeit)[,])
Informiert über alle Dateien, deren DELETION-DATE und Freigabezeit größer oder gleich dem angegebenen Zeitpunkt ist. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (,datum(zeit))
Informiert über alle Dateien, deren DELETION-DATE und Freigabezeit kleiner oder gleich dem angegebenen Zeitpunkt ist. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (datum1(zeit),datum2(zeit))
Informiert über alle Dateien, deren DELETION-DATE innerhalb des angegebenen Zeitraums liegt (datum1 ≤ Freigabedatum ≤ datum2). Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.

DISKWR
Der Anwender kann die zu bearbeitenden Dateien auswählen abhängig von dem im Katalog vereinbarten Zeitpunkt, zu dem Datenkonsistenz gefordert wird.

= ANY
Der im Katalog vereinbarten Zeitpunkt, zu dem Datenkonsistenz gefordert wird, ist kein Auswahlkriterium.

= IMMEDIATE
Informiert über alle Dateien, bei denen Datenkonsistenz direkt nach Beendigung einer Schreiboperation gefordert wird. Diese Dateien eignen sich nicht zur Bearbeitung in einem Schreib-Cache.

= BY-CLOSE
Informiert über alle Dateien, bei denen Datenkonsistenz erst nach der CLOSE-Verarbeitung gefordert wird. Diese Dateien eignen sich zur Bearbeitung in einem Schreib-Cache.

ENCRYPT
Der Anwender kann Dateien danach auswählen, ob und mit welcher Verschlüsselungsmethode sie verschlüsselt sind.

= *ANY
Es sollen alle Dateien bearbeitet werden, unabhängig davon, ob und mit welcher Verschlüsselungsmethode sie verschlüsselt sind.

= *NONE
Es werden nur die Dateien, die nicht verschlüsselt sind, selektiert.

= *AES
Es werden nur die Dateien, die mit der AES-Verschlüsselungsmethode verschlüsselt sind, selektiert.

= * DES
Es werden nur die Dateien, die mit der DES-Verschlüsselungsmethode verschlüsselt sind, selektiert.

= (list_of_encrypt)
Es werden nur Dateien bearbeitet, die einem der angegebenen Auswahlkriterien entsprechen. Innerhalb der Liste können alle Werte außer ANY angegeben werden.

EXDATE
Der Anwender kann über das Freigabedatum (Expiration Date) die Dateien auswählen, die bearbeitet werden sollen.
Das im Katalog vereinbarte Freigabedatum gibt an, ab wann die Datei erstmals wieder verändert oder gelöscht werden darf. Wird beim Erstellen der Datei kein Freigabedatum vereinbart, erhält es denselben Wert wie das Erstellungsdatum.

Datumsangaben kann der Anwender durch eine Zeitangabe ergänzen. Hierbei ist zu beachten, dass derzeit als Freigabezeitpunkt immer die Uhrzeit 00:00:00 im Dateikatalog eingetragen ist.
Die Regeln für die Datums- und Zeitangaben sind auf Seite 868 beschrieben. Bereichsangaben gelten jeweils einschließlich der angegebenen Grenzen.

= ANY
Das Freigabedatum ist kein Auswahlkriterium.

= NONE
Informiert über alle Dateien, für die noch kein Freigabedatum im Katalog eingetragen ist, d.h. Dateien, die noch nicht eröffnet wurden.

= datum
Informiert über alle Dateien, für die das angegebene Freigabedatum vereinbart ist.

= (datum[,])
Informiert über alle Dateien, deren Freigabedatum größer oder gleich dem angegebenen Datum ist.

= (,datum)
Informiert über alle Dateien, deren Freigabedatum kleiner oder gleich dem angegebenen Datum ist.

= (datum1,datum2)
Informiert über alle Dateien, deren Freigabedatum innerhalb des angegebenen Zeitraums liegt (datum1 ≤ Freigabedatum ≤ datum2).

= datum(zeit[,])
Informiert über alle Dateien, für die das angegebene Freigabedatum vereinbart ist und die Uhrzeit der Freigabe größer oder gleich der angegebenen Zeit ist. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= datum(zeit1,zeit2)
Informiert über alle Dateien, für die das angegebene Freigabedatum vereinbart ist und die Uhrzeit der Freigabe innerhalb des angegebenen Zeitintervall liegt. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (datum(zeit)[,])
Informiert über alle Dateien, deren Freigabedatum und Freigabezeit größer oder gleich dem angegebenen Zeitpunkt ist. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (,datum(zeit))
Informiert über alle Dateien, deren Freigabedatum und Freigabezeit kleiner oder gleich dem angegebenen Zeitpunkt ist. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (datum1(zeit),datum2(zeit))
Informiert über alle Dateien, deren Freigabedatum innerhalb des angegebenen Zeitraums liegt (datum1 ≤ Freigabedatum ≤ datum2). Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.

EXTENTS
Nur für Dateien auf Platte oder Net-Storage:
Informiert über alle Dateien entsprechend der angegebenen Zahl ihrer Extents. Ein Extent ist ein zusammenhängender Bereich, den eine Datei auf einer Platte belegt. Die Anzahl der Extents, aus der eine Datei besteht, ist im Katalog hinterlegt.

Für „zahl“ gilt: 0 ≤ zahl ≤ 65535; Bereichsangaben gelten jeweils einschließlich der Bereichsgrenzen.

= ANY
Die Anzahl der Extents ist kein Auswahlkriterium.

= zahl
Es werden nur Plattendateien mit genau der angegebenen Zahl von Extents bearbeitet.

= (zahl [,])
Es werden nur Plattendateien bearbeitet, die mindestens so viele Extents haben wie angegeben (Anzahl der Extents ≥ zahl).

= (,zahl)
Es werden nur Plattendateien bearbeitet, die höchstens so viele Extents haben wie angegeben (Anzahl der Extents ≤ zahl).

= (zahl1, zahl2)
Es werden nur Plattendateien bearbeitet, die mindestens so viele Extents haben wie „zahl1“ und höchstens so viele wie „zahl2“ (zahl1 ≤ Anzahl der Extents ≤ zahl2).

FCBTYPE
Informiert über Dateien/Dateigenerationen in Abhängigkeit von der Zugriffsmethode, mit der sie erstellt wurden. Werden mehrere Zugriffsmethoden in Listenform angegeben, nimmt das System eine logische Oder-Verknüpfung vor und informiert über alle Dateien, die einer der Bedingungen genügen.

= ANY
Die Zugriffsmethode ist kein Auswahlkriterium.

= NONE
Informiert über Dateien, die zwar katalogisiert sind, aber keine Daten enthalten, d.h. die noch nicht eröffnet wurden oder deren Speicherplatz freigegeben wurde mit ERASE..., SPACE.

= ISAM
Informiert über ISAM-Dateien.

= BTAM
Informiert über BTAM-Dateien.

= SAM
Informiert über SAM-Dateien.

= PAM
Informiert über PAM-Dateien.

= (list-of-fcbtype)
In einer Liste können mehrere Zugriffsmethoden angegeben werden. Informiert über alle Dateien, die mit einer der angegebenen Zugriffsmethoden erstellt wurden.

FILTYPE
Informiert über alle Dateien, die dem angegebenen Dateityp entsprechen.

= *ANY
Informiert sowohl über BS2000-Dateien als auch über Node-Files.

= *BS2000
Informiert nur über BS2000-Dateien.

= *NODE
Informiert nur über Node-Files (Dateien, die von BS2000 und von offenen Systemen angelegt und geändert werden können).

FROM
Der Operand FROM definiert die Quelle für die FSTAT-Informationen.

= CATALOG
Der FSTAT-Makroaufruf bezieht seine Informationen aus dem Katalog des Default-Pubsets der Benutzerkennung, d.h. dem Katalog mit der Default-Catid.

= LOCALPVS
Der FSTAT-Makroaufruf bezieht seine Informationen aus den Systemkatalogen aller selektierten lokalen Pubsets.

= (vsn,gerät)
Der FSTAT-Makroaufruf bezieht seine Informationen aus dem Inhaltsverzeichnis der mit „vsn“ bezeichneten Privatplatte bzw. aus dem Katalog des mit „vsn“ bezeichneten Net-Storage-Volumes.
Für ein Net-Storage-Volume ist bei „gerät“ der Volumetyp NETSTOR anzugeben. Andernfalls muss der Gerätetyp der Privatplatte angegeben werden; mögliche Werte sind der Gerätetabelle im Handbuch „Systeminstallation [16] zu entnehmen. Die Operanden „VOLUME“, „SUPPORT“ und „VTOC“ dürfen nicht angegeben werden.

FSIZE
Nur für Plattendateien:
Informiert über Dateien/Dateigenerationsgruppen in Abhängigkeit von der Größe freien (= reservierten, aber nicht belegten) Speicherplatzes; $0 \leq$ zahl ≤ 16777215

= ANY
Die Größe des freien (= reservierten, aber nicht belegten) Speicherplatzes ist kein Auswahlkriterium.

= SIZE
Informiert über Dateien, bei denen die Zahl der freien PAM-Seiten gleich der der reservierten ist.

= zahl
Informiert über Dateien mit genau der angegebenen Anzahl reservierter, aber nicht belegter PAM-Seiten.

= (zahl[,])
Informiert über Dateien mit mindestens der angegebenen Zahl reservierter, aber nicht belegter PAM-Seiten.

= (,zahl)
Informiert über Dateien mit höchstens der angegebenen Zahl reservierter, aber nicht belegter PAM-Seiten.

= (zahl1,zahl2)
Informiert über Dateien, deren Anzahl freier Seiten im angegebenen Bereich liegt (zahl1 < zahl2).

GEN
Der Operand „GEN“ legt fest, ob Informationen zu Dateigenerationen ausgegeben werden.

Die Wechselwirkung mit der Angabe TYPE=FGG gibt die folgende Tabelle wieder:

Operandenkombinationen			Informationen zu		
TYPE=FGG	**GEN=YES**	**GEN=NO**	**FGG**	**Dateigenerationen**	**Dateien**
x	x		*	*	-
x		x	*	-	-
	x		*	*	*
		x	*	-	*

x Angabe im FSTAT-Makroaufruf

* bei der Makrobearbeitung berücksichtigt

\- bei der Makrobearbeitung nicht berücksichtigt

= NO
Es werden keine Informationen zu Dateigenerationen ausgegeben.

= YES
Es werden Informationen zu Dateigenerationen ausgegeben.

Die Angabe GEN=YES wird nur berücksichtigt, wenn in „pfadname“ kein „dateiname“ angegeben wurde.

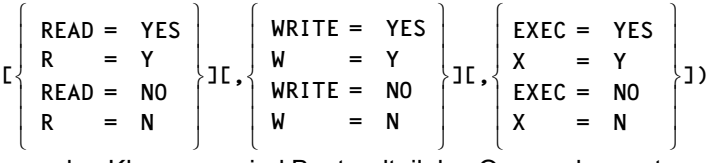
GROUPAR
Informiert über Dateien in Abhängigkeit von den Zugriffsrechten, die in ihren BASIC-ACL-Einträgen für die Mitglieder der Benutzergruppe des Dateieigentümers festgelegt sind.

= ANY
Die BASIC-ACL-Einträge für die Mitglieder der Benutzergruppe des Dateieigentümer sind kein Auswahlkriterium.

= NO-ACCESS
Informiert über alle Dateien, auf die die Benutzergruppe des Eigentümers nicht zugreifen darf.

= zugriffsliste
Informiert über alle Dateien, in deren BASIC-ACL-Eintrag für die Benutzergruppe des Dateieigentümers mindestens eines der in der Liste angegebenen Zugriffsrechte vereinbart ist.

zugriffsliste hat folgendes Format:



Die runden Klammern sind Bestandteil des Operandenwertes und müssen mit angegeben werden.

Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=YES bzw. R=Y	Informiert über alle Dateien, auf die die Benutzergruppe des Eigentümers lesend zugreifen darf.
READ=NO bzw. R=N	Informiert über alle Dateien, auf die die Benutzergruppe des Eigentümers nicht lesend zugreifen darf.
WRITE=YES bzw. W=Y	Informiert über alle Dateien, auf die die Benutzergruppe des Eigentümers schreibend zugreifen darf.
WRITE=NO bzw. W=N	Informiert über alle Dateien, auf die die Benutzergruppe des Eigentümers nicht schreibend zugreifen darf.
EXEC=YES bzw. X=Y	Informiert über alle Dateien, die die Benutzergruppe des Eigentümers ausführen darf.
EXEC=NO bzw. X=N	Informiert über alle Dateien, die die Benutzergruppe des Eigentümers nicht ausführen darf.

GUARDS
Der Anwender kann die zu bearbeitenden Dateien auswählen abhängig von einem vereinbarten Zugriffsschutz mit GUARDS (siehe Handbuch „SECOS“ [8]).

= *ANY
Der vereinbarte Zugriffsschutz mit GUARDS ist kein Auswahlkriterium.

= *NONE
Informiert über alle Dateien, die keinen Zugriffsschutz über GUARDS definiert haben.

= *YES
Informiert über alle Dateien, die einen Zugriffsschutz über GUARDS definiert haben.

= (READ=...,WRITE=...,EXEC=...)
Innerhalb einer Liste kann der Anwender angeben, wie der Zugriffschutz mit GUARDS für die auszuwählenden Dateien vereinbart sein soll. Für jede Zugriffsart (Lesen, Schreiben und Ausführen) kann der vereinbarte Schutz genau angegeben werden. Wird für eine Zugriffsart keine Angabe gemacht, so erfolgt die Auswahl unabhängig von dem dafür vereinbarten Schutz.

Je Zugriffsart kann einer der folgenden Werte angegeben werden:

*ANY	Der vereinbarte GUARDS-Schutz ist kein Auswahlkriterium.
*NONE	Für die angegebene Zugriffsart ist kein Guard vereinbart, d.h. der entsprechende Zugriff wird untersagt.
fname	Für die angegebene Zugriffsart sind im Guard fname alle Bedingungen für die Zugriffserlaubnis enthalten.

IOPERF
Performance-Eigenschaft der Datei. Informiert über alle Dateien, abhängig von dem Performance-Attribut, das im Katalog vereinbart wurde (siehe Operand IOPERF, CATAL-Makro).

= ANY
Die Performance-Eigenschaft ist kein Auswahlkriterium.

= STD
Informiert über alle Dateien, deren Performance-Attribut mit STD vereinbart wurde.

= HIGH
Informiert über alle Dateien, deren Performance-Attribut mit HIGH vereinbart wurde (hohe Performance-Priorität).

= VERY-HIGH
Informiert über alle Dateien, deren Performance-Attribut mit VERY-HIGH vereinbart wurde (höchste Performance-Priorität).

= (list-of-ioperf)
In einer Liste können bis zu drei Performance-Attribute angegeben werden. Informiert über alle Dateien, die eines der angegebenen Attribute besitzen.

IOUSAGE
Informiert über alle Dateien, abhängig von der Art der Ein/Ausgabe-Operationen, auf die sich das Performance-Attribut bezieht (siehe Operand IOUSAGE, CATAL-Makro).

= ANY
Das Performance-Attribut ist kein Auswahlkriterium.

= RDWRT
Informiert über alle Dateien, deren Performance-Attribut sich auf Schreib- und Leseoperationen bezieht.

= WRITE
Informiert über alle Dateien, deren Performance-Attribut sich nur auf Schreiboperationen bezieht.

= READ
Informiert über alle Dateien, deren Performance-Attribut sich nur auf Leseoperationen bezieht.

= (list-of-iousage)
In einer Liste können mehrere Arten von Ein/Ausgabe-Operationen angegeben werden. Informiert über alle Dateien, deren Performance-Attribut sich auf eine der angegebenen Ein/Ausgabe-Operationen bezieht.

LADATE
Informiert über alle Dateien mit entsprechendem Datum des letzten Zugriffs.

Datumsangaben kann der Anwender durch eine Zeitangabe ergänzen. Die Regeln für die Datums- und Zeitangaben sind auf Seite 868 beschrieben.
Bereichsangaben gelten jeweils einschließlich der angegebenen Grenzen.

= ANY
Das Datum des letzten Zugriffs ist kein Auswahlkriterium.

= NONE
Informiert über alle Dateien, für die noch kein Zugriffsdatum im Katalog eingetragen ist, d.h. Dateien, die noch nicht eröffnet wurden.

= datum
Informiert über alle Dateien, auf die an dem angegebenen Tag zuletzt zugegriffen wurde.

= (datum[,])
Informiert über alle Dateien, auf die ab dem angegebenen Datum zuletzt zugegriffen wurde (letzter Zugriff ≥ datum).

= (,datum)
Informiert über alle Dateien, auf die bis zu dem angegebenen Datum zuletzt zugegriffen wurde (letzter Zugriff ≤ datum).

= (datum1,datum2)
Informiert über alle Dateien, auf die während des angegebenen Zeitraums zuletzt zugegriffen wurde (datum1 ≤ letzter Zugriff ≤ datum2).

= datum(zeit[,])
Informiert über alle Dateien, auf die an dem angegebenen Tag und ab der angegebenen Uhrzeit zuletzt zugegriffen wurde.

= datum(zeit1,zeit2)
Informiert über alle Dateien, auf die an dem angegebenen Tag und innerhalb des angegebenen Zeitraums zuletzt zugegriffen wurde.

= (datum(zeit)[,])
Informiert über alle Dateien, auf die ab dem angegebenen Tag und ab der angegebenen Uhrzeit zuletzt zugegriffen wurde.

= (,datum(zeit))
Informiert über alle Dateien, auf die vor dem angegebenen Tag und der angegebenen Uhrzeit zuletzt zugegriffen wurde.

= (datum1(zeit),datum2(zeit))
Informiert über alle Dateien, auf die in dem angegebenen Zeitraum zuletzt zugegriffen wurde. Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.

LASTPAG
Informiert über Dateien in Abhängigkeit vom belegten Speicherplatz (d.h. von der Anzahl beschriebener PAM-Seiten).

= ANY
Der belegte Speicherplatz ist kein Auswahlkriterium.

= zahl
Informiert über alle Dateien mit genau der angegebenen Anzahl beschriebener PAM-Seiten.

= (zahl,)
Informiert über alle Dateien mit mindestens der angegebenen Anzahl beschriebener PAM-Seiten.

= (,zahl)
Informiert über alle Dateien mit höchstens der angegebenen Anzahl beschriebener PAM-Seiten.

= (zahl1,zahl2)
Informiert über alle Dateien, deren Anzahl beschriebener PAM-Seiten im angegebenen Bereich liegt (zahl1 < zahl2).

LCDATE
Informiert über alle Dateien entsprechend dem Datum des letzten Schreibzugriffs.

Datumsangaben kann der Anwender durch eine Zeitangabe ergänzen. Die Regeln für die Datums- und Zeitangaben sind auf Seite 868 beschrieben.
Bereichsangaben gelten jeweils einschließlich der angegebenen Grenzen.

= ANY
Das Datum des letzten Schreibzugriffs ist kein Auswahlkriterium.

= NONE
Informiert über alle Dateien, für die noch kein Datum des letzten Schreibzugriffs im Katalog eingetragen ist, d.h. Dateien, die noch nicht eröffnet wurden.

= datum
Informiert über alle Dateien, auf die an dem angegebenen Tag zuletzt schreibend zugegriffen wurde.

= (datum[,])
Informiert über alle Dateien, auf die ab dem angegebenen Datum zuletzt schreibend zugegriffen wurde (letzter Zugriff ≥ datum).

= (,datum)
Informiert über alle Dateien, auf die bis zu dem angegebenen Datum zuletzt schreibend zugegriffen wurde (letzter Zugriff ≤ datum).

= (datum1,datum2)
Informiert über alle Dateien, auf die während des angegebenen Zeitraums zuletzt schreibend zugegriffen wurde (datum1 ≤ letzter Zugriff ≤ datum2).

= datum(zeit[,])
Informiert über alle Dateien, auf die an dem angegebenen Tag und ab der angegebenen Uhrzeit zuletzt schreibend zugegriffen wurde.

= datum(zeit1,zeit2)
Informiert über alle Dateien, auf die an dem angegebenen Tag und innerhalb des angegebenen Zeitraums zuletzt schreibend zugegriffen wurde.

= (datum(zeit)[,])
Informiert über alle Dateien, auf die ab dem angegebenen Tag und ab der angegebenen Uhrzeit zuletzt schreibend zugegriffen wurde.

= (,datum(zeit))
Informiert über alle Dateien, auf die vor dem angegebenen Tag und der angegebenen Uhrzeit zuletzt schreibend zugegriffen wurde.

= (datum1(zeit),datum2(zeit))
Informiert über alle Dateien, auf die in dem angegebenen Zeitraum zuletzt schreibend zugegriffen wurde. Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.

MANCLAS
Informiert über alle Dateien entsprechend der HSMS-Management-Klasse zur Dateisicherung auf SM-Pubsets.

= *ANY
Die HSMS-Management-Klasse ist kein Auswahlkriterium.

= *NONE
Nur Dateien, für die keine HSMS-Management-Klasse definiert ist, werden ausgewählt.

= <c-string 1..8>
Nur Dateien mit der angegebenen HSMS-Management-Klasse werden ausgewählt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben. In allen Makroaufrufen, die sich durch den MF-Operanden unterscheiden (MF=L/E/D/...), muss der Operand VERSION den gleichen Wert haben.

Besonderheit bei MF=D/C:
Über den Operanden CEINFO kann der Benutzer steuern, welche Ausgabeblöcke generiert werden sollen (XPAND=OUTPUT). Es werden nur die Blöcke bei MF=D/C expandiert, die der Benutzer bei CEINFO spezifiziert hat.

MIGRATE
Informiert über alle Dateien, die im Katalog den angegebenen Eintrag für MIGRATE besitzen. Dieser Eintrag wird vom Hierarchischen Speicher Management System HSMS bei der Verdrängung (Migration) von Dateien ausgewertet (siehe Makro CATAL, Operand MIGRATE Seite 170). Bei Angabe in Listenform werden alle Dateien selektiert, die einer der Bedingungen genügen.

= ANY
Der MIGRATE-Eintrag ist kein Auswahlkriterium.

= ALLOWED
Informiert nur über Dateien, für die im Katalogeintrag MIGRATE=ALLOWED vereinbart wurde, d.h. Dateien, die auf die Speicherebene S1 oder S2 verdrängt werden dürfen.

= INHIBIT
Informiert nur über Dateien, für die im Katalogeintrag MIGRATE=INHIBIT vereinbart wurde, d.h. Dateien, die kurzfristig (z.B. für Reorganisationszwecke) verdrängt werden dürfen.

= FORBIDDEN
Informiert nur über Dateien, für die im Katalogeintrag MIGRATE=FORBIDDEN vereinbart wurde, d.h. Dateien, die nicht verdrängt werden dürfen.

= (list-of-migrate)
Der Anwender kann die gewünschten Werte in einer Liste angeben. Informiert über alle Dateien, für die im Katalog einer der angegebenen Werte vereinbart wurde.

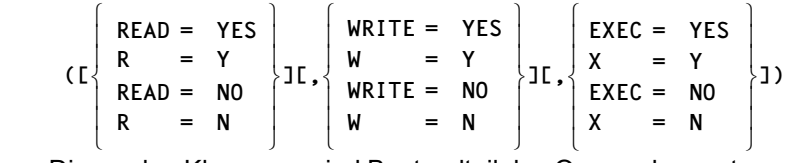
OTHERAR
Informiert über Dateien in Abhängigkeit von den Zugriffsrechten, die in ihren BASIC-ACL-Einträgen für alle Anwender außerhalb der Benutzergruppe des Dateieigentümers festgelegt sind.

= <u>ANY</u>
Die BASIC-ACL-Einträge für alle Anwender außerhalb der Benutzergruppe des Dateieigentümers sind kein Auswahlkriterium.

= NO-ACCESS
Informiert über alle Dateien, auf die Anwender außerhalb der Benutzergruppe des Eigentümers nicht zugreifen dürfen.

= zugriffsliste
Informiert über alle Dateien, in deren BASIC-ACL-Eintrag für Anwender außerhalb der Benutzergruppe des Dateieigentümers mindestens eines der in der Liste angegebenen Zugriffsrechte vereinbart ist.

zugriffsliste hat folgendes Format:



Die runden Klammern sind Bestandteil des Operandenwertes und müssen mit angegeben werden.

Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=YES bzw. R=Y	Informiert über alle Dateien, auf die Anwender außerhalb der Benutzergruppe des Eigentümers lesend zugreifen dürfen.
READ=NO bzw. R=N	Informiert über alle Dateien, auf die Anwender außerhalb der Benutzergruppe des Eigentümers nicht lesend zugreifen dürfen.
WRITE=YES bzw. W=Y	Informiert über alle Dateien, auf die Anwender außerhalb der Benutzergruppe des Eigentümers schreibend zugreifen dürfen.
WRITE=NO bzw. W=N	Informiert über alle Dateien, auf die Anwender außerhalb der Benutzergruppe des Eigentümers nicht schreibend zugreifen dürfen.
EXEC=YES bzw. X=Y	Informiert über alle Dateien, die Anwender außerhalb der Benutzergruppe des Eigentümers ausführen dürfen.
EXEC=NO bzw. X=N	Informiert über alle Dateien, die Anwender außerhalb der Benutzergruppe des Eigentümers nicht ausführen dürfen.

OUTAREA
Ausgabebereich für Informationen aus dem Katalogeintrag.

= (<list-of-elements-002>)
Der Ausgabebereich wird in Listenform angegeben, bestehend aus:
– Adresse des Ausgabebereichs. Sie kann nur als Konstante oder als Equate angegeben werden und muss auf Wortgrenze ausgerichtet sein.
– Länge des Ausgabebereiches (in Byte). Welcher Wert anzugeben ist, hängt vom Informationsumfang ab, der über den Operanden OUTPUT angefordert wird.
Sie kann nur als Konstante oder als Equate angegeben werden.

Das Format des Ausgabebereiches ist auf Seite 655 beschrieben.

Hinweis
Für die beiden Ausgabebereiche (OUTAREA, STOUTAR) werden intern SPECIFIED-Bits gesetzt. Wenn eine Ausgabe erfolgen soll, muss das jeweilige AREA-SPECIFIED-Bit gesetzt sein. Die Bits werden bei MF=L gesetzt. Aus diesem Grund ist es sinnvoll, beim MF=L-Aufruf den Operand OUTAREA bzw. STOUTAR mit Dummy-Werten zu versorgen.

OUTPUT
Die Ausgabestruktur in Blöcken besteht aus zwei unabhängigen Teilen: Informationen aus dem Katalogeintrag und statistische Informationen über alle selektierten Katalogeinträge. Zum Aufbau der verschiedenen Informationsblöcke siehe Seite 655.

Die Informationen aus dem Katalogeintrag werden in den durch den Operanden OUTAREA spezifizierten Ausgabebereich übertragen.

Die statistischen Informationen werden in den durch den Operanden STOUTAR spezifizierten Ausgabebereich übertragen.

= RC-ONLY
Es wird nur der Returncode zurückgegeben. Ein Ausgabebereich wird nicht benötigt.

= FNAM-ONLY
Es werden nur die Pfadnamen (mit PVSID, USERID und FILENAME) in den Ausgabebereich übertragen (Beschreibung siehe Seite 657).

Die Pfadnamen werden in den durch den Operanden OUTAREA spezifizierten Ausgabebereich übertragen.

= CEINFO
Es werden Informationen aus den Katalogeinträgen ausgegeben. Welche Informationen ausgegeben werden, kann über den Operanden CEINFO gesteuert werden. Statistische Informationen werden nicht gegeben (Beschreibung siehe Seite 655).

Die Informationen aus dem Katalogeintrag werden in den durch den Operanden OUTAREA spezifizierten Ausgabebereich übertragen.

= STAT-SHORT
Es werden nur die statistischen Informationen in den statistischen Ausgabebereich übertragen (Beschreibung siehe Seite 660).

Die statistischen Informationen werden in den durch den Operanden STOUTAR spezifizierten Ausgabebereich übertragen.

Hinweis
Es können keine statistischen Daten übertragen werden, wenn gleichzeitig der Operand FROM=(volume,device) spezifiziert wird. Ein derartiger Funktionsaufruf wird mit Returncode DMS0576 abgewiesen.

= STAT-LONG
Es werden nur die statistischen Informationen in den statistischen Ausgabebereich übertragen. Zusätzlich zur Ausgabe von STAT-SHORT werden auch statistische Informationen pro PVSID und USERID übertragen (Beschreibung siehe Seite 658).

Die statistischen Informationen werden in den durch den Operanden STOUTAR spezifizierten Ausgabebereich übertragen.

Hinweis

Es können keine statistischen Daten übertragen werden, wenn gleichzeitig der Operand FROM=(volume,device) spezifiziert wird. Ein derartiger Funktionsaufruf wird mit Returncode DMS0576 abgewiesen.

= STAT-INFO
Es werden die Informationen der Katalogeinträge und die statistischen Informationen ausgegeben (entspricht der Ausgabemenge CEINFO + STAT-LONG).

Beide Teile der Ausgabe werden mit Pointern miteinander verkettet (Beschreibung siehe Seite 661).

Die Informationen aus dem Katalogeintrag werden in den durch den Operanden OUTAREA spezifizierten Ausgabebereich übertragen.

Die statistischen Informationen werden in den durch den Operanden STOUTAR spezifizierten Ausgabebereich übertragen.

Hinweis

Es können keine statistischen Daten übertragen werden, wenn gleichzeitig der Operand FROM=(volume,device) spezifiziert wird. Ein derartiger Funktionsaufruf wird mit Returncode DMS0576 abgewiesen.

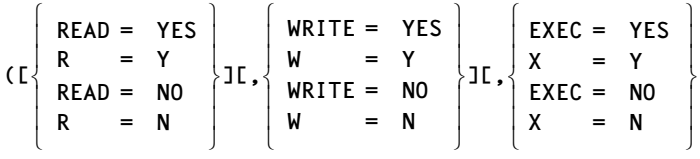
OWNERAR
Informiert über Dateien in Abhängigkeit von den Zugriffsrechten, die in ihren BASIC-ACL-Einträgen für die Dateieigentümer festgelegt sind.

= ANY
Die BASIC-ACL-Einträge für die Dateieigentümer sind kein Auswahlkriterium.

= NO-ACCESS
Informiert über alle Dateien, auf die der Eigentümer nicht zugreifen darf.

= zugriffsliste
Informiert über alle Dateien, in deren BASIC-ACL-Eintrag für den Dateieigentümer mindestens eines der in der Liste angegebenen Zugriffsrechte vereinbart ist.

zugriffsliste hat folgendes Format:



Die runden Klammern sind Bestandteil des Operandenwertes und müssen mit angegeben werden.

Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=YES bzw. R=Y	Informiert über alle Dateien, auf die der Eigentümer lesend zugreifen darf.
READ=NO bzw. R=N	Informiert über alle Dateien, auf die der Eigentümer nicht lesend zugreifen darf.
WRITE=YES bzw. W=Y	Informiert über alle Dateien, auf die der Eigentümer schreibend zugreifen darf.
WRITE=NO bzw. W=N	Informiert über alle Dateien, auf die der Eigentümer nicht schreibend zugreifen darf.
EXEC=YES bzw. X=Y	Informiert über alle Dateien, die der Eigentümer ausführen darf.
EXEC=NO bzw. X=N	Informiert über alle Dateien, die der Eigentümer nicht ausführen darf.

PASS
Informiert über Dateien/Dateigenerationsgruppen in Abhängigkeit von einem mit CATAL definierten Kennwortschutz. Werden mehrere Kennwortarten in Listenform angegeben, nimmt das System eine logische Oder-Verknüpfung vor und informiert über alle Dateien, die einer der genannten Bedingungen genügen. Kennwörter werden nicht ausgegeben.

= ANY
Der Kennwortschutz ist kein Auswahlkriterium.

= NONE
Informiert über Dateien, für die kein Kennwortschutz besteht.

= RDPASS
Informiert darüber, welche Dateien durch ein Lesekennwort geschützt sind.

= WRPASS
Informiert darüber, welche Dateien durch ein Schreibkennwort geschützt sind.

= EXPASS
Informiert darüber, welche Dateien mit einem Ausführungskennwort geschützt sind.

= (list-of-pass)
In einer Liste kann der Anwender mehrere Arten des Kennwortschutzes angeben. Informiert über alle Dateien, die mit einem der angegebenen Kennworttypen geschützt sind.

PASSW
Der Operand PASSW bestimmt, wie Kennwörter in den Ausgabebereich übertragen werden.

= NO
Kennwörter werden im Ausgabebereich nicht explizit dargestellt. Die entsprechenden Felder sind binär 0 gesetzt.

= YES
Nur für privilegierte Anwender
Kennwörter werden im Ausgabebereich explizit dargestellt.

PREFIX
wird nur in Verbindung mit MF=C oder MF=D ausgewertet und legt fest, mit welchen Zeichen die Feldnamen und Equates beginnen, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein 1-3 Zeichen langes Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen sollen.

PREFORM
Informiert über Dateien abhängig von deren (beabsichtigten) Dateiformaten auf SM-Pubsets.

= *ANY
Das Dateiformat ist kein Auswahlkriterium.

= *NONE
Informiert über alle Dateien, für die kein PREFORMAT-Wert definiert wurde.

= *K
Informiert über alle Dateien, die das beabsichtigte Dateiformat *K besitzen.

= *NK2
Informiert über alle Dateien, die das beabsichtigte Dateiformat *NK2 besitzen.

= *NK4
Informiert über alle Dateien, die das beabsichtigte Dateiformat *NK4 besitzen.

= (list-of-preform)
Informiert über alle Dateien, die einem der angegebenen Dateiformate entsprechen. Innerhalb der Liste können alle Werte außer ANY angegeben werden.

PROTACT
Informiert über die Dateien abhängig von der Schutzstufe der höchsten aktivierten Zugriffskontrolle.

Für Zugriffe auf die Datei gilt der höchste aktivierte Zugriffsschutz. Die nachfolgende Tabelle zeigt Art der Zugriffskontrolle, Schutzmerkmal, das im CATAL-Makroaufruf anzugeben ist und die Rangfolge (Schutzstufe):

Zugriffsschutz	Schutzmerkmal	Schutzstufe
Standard-Zugriffskontrolle	ACCESS u. SHARE	0
Einfache Zugriffskontroll-Liste	BASACL, OWNERAR, GROUPAR, OTHERAR	1
Zugriffskontrolle über GUARDS	GUARDS	2

Alle weiteren Schutzmerkmale der Datei (z.B. Kennwörter) werden unabhängig von der realisierten Schutzstufe ausgewertet.

= ANY
Informiert über alle Dateien, unabhängig von der Schutzstufe der höchsten aktivierten Zugriffskontrolle.

= LEVEL-0
Informiert über alle Dateien, bei denen die Zugriffe über die Standard-Zugriffskontrolle erfolgen.

= LEVEL-1
Informiert über alle Dateien, bei denen die Zugriffe über eine einfache Zugriffskontroll-Liste (BASIC-ACL-Schutz) erfolgen.

= LEVEL-2
Informiert über alle Dateien, bei denen die Zugriffe über GUARDS erfolgen.

= (list-of-protact)
Der Anwender kann in einer Liste maximal 3 Schutzstufen angeben. Informiert über alle Dateien, bei denen die Zugriffe über eine Zugriffskontrolle erfolgen, die einer der angegebenen Schutzstufen entspricht.

RELSPAC
Informiert über alle Dateien, abhängig von einer vereinbarten Sperre gegen Freigabe von nicht belegtem Speicherplatz mit dem FILE-Makroaufruf bzw. dem Kommando MODIFY-FILE-ATTRIBUTES. Die Sperre kann mit dem CATAL-Makro im Katalog vereinbart werden.

= ANY
Die Sperre gegen Freigabe von nicht belegtem Speicherplatz ist kein Auswahlkriterium.

= ALLOWED
Es werden alle Dateien bearbeitet, bei denen nicht belegter Speicherplatz freigegeben werden darf.

= IGNORED
Es werden alle Dateien bearbeitet, bei denen nicht belegter Speicherplatz nicht freigegeben werden darf.

SHARE
Informiert über Dateien/Dateigenerationsgruppen in Abhängigkeit davon, ob sie mehrbenutzbar sind. Ist mit „$userid.“ eine fremde Benutzerkennung, die nicht Miteigentümer ist, angegeben, gilt immer implizit SHARE=YES. Der Anwender kann die zu bearbeitenden Dateien auswählen abhängig von ihrer Mehrbenutzbarkeit (siehe Operand SHARE im CATAL-Makro).

= ANY
Die Mehrbenutzbarkeit ist kein Auswahlkriterium.

= YES
Informiert über alle Dateien, die mehrbenutzbar sind, d.h. Dateien, die bei aktiver Standard-Zugriffskontrolle auch für fremde Benutzerkennungen zugreifbar sind.

= NO
Informiert über alle Dateien, die nicht mehrbenutzbar sind, d.h. Dateien, die bei aktiver Standard-Zugriffskontrolle nur für den Dateieigentümer zugreifbar sind.

= SPECIAL
Informiert über alle Dateien, die auch für die Kennung mit dem Privileg Hardware-Maintenance zugreifbar sind.

= (list-of-share)
In einer Liste können mehrere Operandenswerte angegeben werden.

SIZE
Informiert über Dateien/Dateigenerationen in Abhängigkeit von der Größe des reservierten Speicherplatzes. Als Werte sind ganze Zahlen von 0 bis 2147483647 erlaubt.

= ANY
Die Größe des reservierten Speicherplatzes ist kein Auswahlkriterium.

= FSIZE
Informiert über Dateien, bei denen die Zahl der reservierten PAM-Seiten gleich der der freien Seiten ist.

= zahl
Informiert über Dateien mit genau der angegebenen Anzahl reservierter PAM-Seiten.

= (zahl[,])
Informiert über Dateien mit min. der angegebenen Anzahl reservierter PAM-Seiten.

= (,zahl)
Informiert über Dateien mit max. der angegebenen Anzahl reservierter PAM-Seiten.

= (zahl1,zahl2)
Informiert über Dateien, deren Anzahl reservierter PAM-Seiten im angegebenen Bereich liegt (zahl1 < zahl2).

SLEVEL
Informiert über Dateien, die sich auf den angegebenen Speicherhierarchie-Ebenen befinden (siehe Handbuch „HSMS“ [10]). HSMS unterstützt die folgenden Speicherhierarchie-Ebenen:

S0: realisiert durch Plattenspeicher mit schnellem Zugriff (Online-Verarbeitung)
S1: realisiert durch Plattenspeicher mit hoher Kapazität (online verfügbare Hintergrundebene)
S2: realisiert durch Magnetband- oder Magnetbandkassettenarchive (offline verfügbare Hintergrundebene)

= ANY
Die Speicherhierarchie-Ebene ist kein Auswahlkriterium.

= S0
Informiert nur über Dateien, die sich auf der Ebene S0 befinden.

= S1
Informiert nur über Dateien, die sich auf der Ebene S1 befinden.

= S2
Informiert nur über Dateien, die sich auf der Ebene S2 befinden.

= (list-of-slevel)
In einer Liste können mehrere Speicherhierarchie-Ebenen angegeben werden.

SORT
Der Operand SORT bestimmt die Sortierung der Katalogeinträge/Pfadnamen in der Ausgabe.

= FILENAM
Die Katalogeinträge/Pfadnamen werden alphabetisch sortiert ausgegeben.

= NO
Die Katalogeinträge/Pfadnamen werden in der Reihenfolge ausgegeben, wie sie im Katalog stehen.

STATE
Informiert über alle Dateien entsprechend ihrem momentanen Bearbeitungszustand.

= ANY
Der momentane Bearbeitungszustand ist kein Auswahlkriterium.

= NOCLOS
Informiert über alle Dateien, die momentan schreibend geöffnet sind. Das können sein:
– normal eröffnete Dateien (Openmodus OUTIN, INOUT, OUTPUT)
– in einer vorherigen Session nicht geschlossene Dateien
– in der laufenden Session nicht geschlossene Dateien, weil der Prozess abgebrochen wurde.

Hinweis

STATE=NOCLOS setzt implizit GEN=YES, d.h. geöffnete Dateigenerationen werden immer mit ausgegeben.

= CLOSED
Informiert über alle Dateien, die bereits geschlossen wurden, d.h. Dateien, die nicht durch NOCLOS ausgewählt werden.

= CACHED
Informiert über alle Dateien, die momentan in einem Cache befinden.

= NOT-CACHED
Informiert über alle Dateien, die momentan nicht über einen Cache verarbeitet werden.

= CACHE-NOT-SAVED
Informiert über alle Dateien, für die es beim Schließen nicht möglich war, alle Schreibdaten vom Cache auf einen Plattenspeicher zu sichern.

= REPAIR-NEEDED
Informiert über alle Dateien, die in einer vorherigen Session nicht geschlossen wurden und auf die noch kein VERIFY durchgeführt wurde (siehe VERIFY-Makro, Seite 853).

= DEFECT-REPORTED
Informiert über alle Dateien, die defekte Plattenblöcke enthalten können.

= NO-OPEN-ALLOWED
Informiert über alle Dateien, die wegen Dateninkonsistenz nicht geöffnet werden können.

= OPEN-ALLOWED
Informiert über alle Dateien, die geöffnet werden können.

= (list-of-state)
Der Anwender kann in einer Liste maximal 7 Dateizustände angeben. Informiert über alle Dateien, die sich in einem der angegebenen Zustände befinden.

STOCLAS
Informiert über alle Dateien entsprechend der Storage-Klasse zur Dateiablage auf SM-Pubsets.

= *ANY
Die Storage-Klasse ist kein Auswahlkriterium.

= *NONE
Nur Dateien, für die keine Storage-Klasse definiert ist, werden ausgewählt.

= <c-string 1..8>
Nur Dateien mit der angegebenen Storage-Klasse werden ausgewählt.

STOTYPE
Informiert über alle Dateien entsprechend des Speichertyps.

= *ANY
Der Speichertyp ist kein Auswahlkriterium.

= *PUBSPACE
Nur Dateien, die auf gemeinschaftlichen Datenträgern liegen, werden ausgewählt.

= *NETSTOR
Nur Dateien, die auf Net-Storage-Volumes liegen, werden ausgewählt.

STOUTAR
Ausgabebereich für statistische Informationen.

= (<list-of-elements-002>)
Der Ausgabebereich wird in Listenform angegeben, bestehend aus:

– Adresse des Ausgabebereichs. Sie kann nur als Konstante oder als Equate angegeben werden und muss auf Wortgrenze ausgerichtet sein.
– Länge des Ausgabebereiches (in Byte). Welcher Wert anzugeben ist, hängt vom Informationsumfang ab, der über den Operanden OUTPUT angefordert wird.
  Sie kann nur als Konstante oder als Equate angegeben werden.

Das Format des Ausgabebereiches ist auf Seite 661 beschrieben.

Hinweis
Für die beiden Ausgabebereiche (OUTAREA, STOUTAR) werden intern SPECIFIED-Bits gesetzt. Wenn eine Ausgabe erfolgen soll, muss das jeweilige AREA-SPECIFIED-Bit gesetzt sein. Die Bits werden bei MF=L bzw. MF=M gesetzt. Aus diesem Grund ist es sinnvoll, beim MF=L-Aufruf den Operand OUTAREA bzw. STOUTAR mit Dummywerten zu versorgen.

SUPPORT
Informiert über Dateien/Dateigenerationen/Dateigenerationsgruppen, abhängig davon, auf welchem Datenträgertyp sie gespeichert sind.

= ANY
Der Datenträgertyp ist kein Auswahlkriterium.

= PUBLIC
Informiert über Dateien usw. auf gemeinschaftlichen Datenträgern oder Net-Storage-Volumes.

= PRDISC
Informiert über Dateien usw. auf Privatplatten.

= TAPE
Informiert über auf Band gespeicherte Dateien, Dateigenerationen oder Dateigenerationsgruppen.

= (list-of-support)
Der Anwender kann in einer Liste maximal 3 Datenträgertypen angeben. Informiert über alle Dateien, die auf einem der angegebenen Datenträgertypen abgespeichert sind.

S0MIGR
Informiert über Dateien abhängig davon, ob eine Umallokierung (Migration) auf S0-Ebene erlaubt ist.

= *ANY
Die Migrations-Erlaubnis ist kein Auswahlkriterium.

= *ALLOWED
Informiert über Dateien, für die eine Migration innerhalb der S0-Ebene erlaubt ist.

= *FORBIDDEN
Informiert über Dateien, für die eine Migration innerhalb der S0-Ebene nicht erlaubt ist.

= (list-of-s0migr)
Der Anwender kann die gewünschten Werte in einer Liste angeben. Informiert über alle Dateien, für die im Katalog einer der angegebenen Werte vereinbart wurde.

TIMBASE
Steuert, ob die absoluten Datumseingaben in UTC- oder lokaler Zeit erfolgen. An diesen Operanden ist auch das Datumsformat der FSTAT-Ausgabe gekoppelt.

= *UTC
Absolute Datumseingaben und alle Datumsausgaben erfolgen in UTC-Zeit.

= *LTI
Absolute Datumseingaben und alle Datumsausgaben erfolgen in lokaler Zeit.

TYPE
Informiert über alle Dateien entsprechend ihrem Typ.

= ANY
Informiert über alle Dateien unabhängig von ihrem Typ.

= FILE
Informiert über alle Dateien mit Ausnahme der Dateigenerationsgruppen.

= FGG
Informiert nur über Dateigenerationsgruppen und bei Angabe von GEN=YES auch über Dateigenerationen (siehe Operand „GEN“ auf Seite 627).

= PLAM
Informiert nur über PLAM-Bibliotheken. Dies ist eine Untermenge der Dateien, die bei der Angabe TYPE=FILE ausgewählt werden.

= (list-of-type)
Informiert über alle Dateien, die den angegebenen Typen entsprechen. Der Anwender kann in einer Liste maximal 3 Typen angeben.

USRINFO
Informiert abhängig von der benutzereigenen Metainformation über Dateien/Dateigenerationen.

= *ANY
Die benutzereigene Metainformation ist kein Auswahlkriterium.

= *NONE
Informiert über Dateien, die keine benutzereigene Metainformation besitzen.

= <c-string 1..8>
Informiert über Dateien mit der angegebenen benutzereigenen Metainformation.

VERSION
Gibt an, welche Version der Parameterliste generiert werden soll.

Hinweis

Wird der FSTAT mit VERSION=0/1 in einer Umgebung aufgerufen, die „große Dateien“ unterstützt, besteht ein Prüfungs- und evtl. ein Umstellungsaufwand. Nähere Informationen dazu finden Sie ab Seite 662.

= 0
Es wird das Parameterlistenformat generiert, das vor BS2000 V8.0A unterstützt wurde. Dieses Format unterstützt allerdings auch nur die bis dahin bekannten Parameter. Z.B. darf der Pfadname nur ohne Musterzeichen angegeben werden und von den Selektionsparametern ist nur VOLUME und POS erlaubt. Die unterstützten Operanden/Operandenwerte können der Tabelle „Versionsunterschiede – VERSION=0/1/2/3/4“ auf Seite 666 entnommen werden.

= 1
Es wird das Parameterlistenformat generiert, das in BS2000 V8.0 bis einschließlich V10.0 unterstützt wurde.
Dieses Format unterstützt allerdings auch nur die bis dahin bekannten Parameter. Die unterstützten Operanden/Operandenwerte können der Tabelle „Versionsunterschiede – VERSION=0/1/2/3/4“ auf Seite 666 entnommen werden.

= 2
Es wird das Parameterlistenformat für Versionen ab BS2000/OSD-BC V1.0 generiert.

= 3
Es wird das Parameterlistenformat für Versionen ab BS2000/OSD-BC V3.0 generiert.

= 4
Es wird das Parameterlistenformat für Versionen ab BS2000/OSD-BC V9.0 generiert.

Hinweis

Wenn schon bestehende Software neu übersetzt werden soll, die Manipulationen an der generierten Parameterliste vornimmt, muss das alte Format angefordert werden. Ansonsten liegt Source-Kompatibilität vor.

VOLSET
Der Anwender kann die zu bearbeitenden Dateien über den Volume-Set auswählen, auf dem sie liegen.

= *ANY
Der Volume-Set ist kein Auswahlkriterium.

= *CONTROL
Informiert über alle Dateien, die auf dem Control-Volume-Set des SM-Pubsets liegen.

= <c-string 1..4>
Informiert über alle Dateien, die auf dem spezifizierten Volume-Set liegen.

VOLUME
Der Anwender kann die zu bearbeitenden Dateien über die Archivnummer (VSN) ihres Datenträgers auswählen.

= ANY
Die Archivnummer (VSN) des Datenträgers ist kein Auswahlkriterium.

= vsn
Informiert über alle Dateien/Dateigenerationsgruppen, die für den Datenträger mit der angegebenen Archivnummer („vsn“) einen Eintrag in ihrer Datenträgerliste enthalten. Bezeichnet „vsn“ keine Privatplatte, so wird über Dateigenerationsgruppen nicht informiert.

VTOC
Der Anwender kann entscheiden, ob die angeforderten Informationen der VTOC (= Volume Table of Contents) einer Privatplatte bzw. eines Net-Storage-Volumes entnommen werden sollen oder dem System-Dateikatalog TSOSCAT. Der Operand VTOC kann nicht auf teilqualifizierte Dateinamen angewendet werden oder in Zusammenhang mit GEN=YES.

= NO
Gibt den aktuellen Eintrag im TSOSCAT aus.

= YES
Gibt die VTOC-Katalogeinträge (aus dem F1-Kennsatz einer privaten Platte bzw. dem Katalog eines Net-Storage-Volumes) aus, entsprechend dem letzten aktuellen Zustand im gesamten Rechnernetz. Der Datenträger muss zugewiesen sein. Der VTOC-Eintrag des Datenträgers ersetzt den entsprechenden TSOSCAT-Eintrag. So kann die Konsistenz zwischen VTOC- und TSOSCAT-Eintrag wiederhergestellt werden, wenn z.B. Dateien einer SPD (siehe „Privatplatten“ im Handbuch „Einführung in das DVS“ [1]) von einem anderen Rechner aus geändert wurden.
Wenn sich die angegebene Datei nicht mehr auf dem im TSOSCAT-Eintrag eingetragenen Datenträger befindet, wird der TSOSCAT-Eintrag gelöscht.

WORKFIL
Informiert über Dateien auf SM-Pubsets, abhängig davon, ob sie vom Systemverwalter gelöscht werden können (Arbeitsdateien).

= *ANY
Es ist kein Auswahlkriterium, ob die Dateien Arbeitsdateien sind oder nicht.

= *NO
Informiert über alle Dateien, die keine Arbeitsdateien sind.

= *YES
Informiert über alle Dateien, die Arbeitsdateien sind.

WTQUIET
Steuert, ob bei einem Shared Pubset bzw. Remote-Imported Pubset, der sich zum Zeitpunkt des FSTAT-Aufrufs bereits im Zustand QUIET befand, auf Beendigung des QUIET-Zustandes gewartet wird. Diese Wartezeit ist mit DIALOG-WAIT-TIME bzw. BATCH-WAIT-TIME im entsprechenden MRSCAT-Eintrag festgelegt.
Andernfalls wird entweder der Auftrag sofort mit dem Returncode DMS0502 abgebrochen oder, bei Wildcards-Angabe in der catid, der Pubset übersprungen.

= *YES
Ist ein Pubset im Zustand QUIET, so führt dies zu einem Wartezustand.

= *NO
Ist ein Pubset im Zustand QUIET, so führt dies zu dem Returncode DMS0502 oder, falls Wildcards in der catid angegeben sind, zum Überspringen des Pubsets.

XPAND
Durch diesen Operanden wird gesteuert, welcher Eingabeparameterbereich (mit oder ohne Selektionsparameterbereich) generiert werden soll. Darüber hinaus können mit XPAND die Datenbeschreibungen (DSECTs) für den Ausgabebereich generiert werden.

= PLSHORT
Der Eingabeparameterbereich wird ohne Selektionsparameterbereich generiert.

= PLLONG
Der Eingabeparameterbereich wird mit Selektionsparameterbereich generiert.

= OUTPUT
Es werden alle Datenbeschreibungen (DSECTs) zur Beschreibung der Ausgabe-Informationsblöcke generiert.

= (PLSHORT,OUTPUT)
Der Eingabeparameterbereich wird ohne Selektionsparameterbereich generiert. Alle Datenbeschreibungen (DSECTs) zur Beschreibung der Ausgabe-Informationsblöcke werden generiert.

= (PLLONG,OUTPUT)
Der Eingabeparameterbereich wird mit Selektionsparameterbereich generiert. Alle Datenbeschreibungen (DSECTs) zur Beschreibung der Ausgabe-Informationsblöcke werden generiert.

Returncodes

Der Fehlercode wird nur noch im Standardheader der Parameterliste und nicht mehr wie in Version 2 im Mehrzweckregister 15 zurückgeliefert.

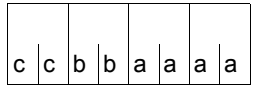
Hinweis

Sind in der catid-Angabe Wildcards vorhanden, so werden folgende Returncodes unterdrückt.

X'02000000'	X'00400503'	X'00400616'	X'00820504'
X'00400501'	X'00400505'	X'00820502'	X'00820506'

Tritt kein anderer Fehler auf und wurde eine Datei selektiert, so ist der Returncode 0, wurde keine Datei selektiert, wird DMS06CC zurückgeliefert.

Standard-header:



Über die Ausführung des Makros FSTAT wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'00'	X'0000'	Kein Fehler
X'02'	X'00'	X'0000'	Ausgabe auf Grund defekter Volume-Sets nicht vollständig; falls IFSVSETI = IFSODVSE (ONE DEFECT VOLUMESET), dann steht in IFSVSET die Kennung des Volume-Sets.
	X'40'	X'0501'	Angeforderter Katalog nicht verfügbar
	X'82'	X'0502'	Angeforderter Katalog im Ruhezustand
	X'40'	X'0503'	Falsche Information im MRSCAT
	X'82'	X'0504'	Fehler im Katalog-Verwaltungs-System
	X'40'	X'0505'	Fehler bei der Rechner-Kommunikation (MRS)
	X'80'	X'0506'	Operation wegen Masterwechsel abgebrochen
	X'40'	X'0510'	Fehler beim Aufruf einer internen Funktion
	X'40'	X'0512'	Angeforderter Katalog unbekannt
	X'40'	X'051B'	Benutzerkennung im angegebenen Pubset unbekannt

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'40'	X'051C'	Kein Zugriffsrecht auf angegebenen Pubset
	X'40'	X'051D'	LOGON-Passwort auf angegebenem Pubset anders
	X'40'	X'052E'	Datenträger nicht mehr verfügbar
	X'20'	X'0530'	Fehler bei der Speicherplatzanforderung
	X'20'	X'0531'	Unerwarteter Fehler beim Katalogzugriff
	X'40'	X'0533'	Angegebene Datei nicht gefunden
	X'82'	X'0534'	Privater Datenträger kann nicht zugewiesen werden
	X'20'	X'053B'	Systemfehler beim Dateizugriff
	X'40'	X'053D'	Katalog oder F1-Etikett-Block ist zerstört
	X'20'	X'054F'	Unerwarteter Fehler beim Zugriff auf JOIN-Datei
	X'82'	X'055A'	Geräte zurzeit belegt
	X'40'	X'055F'	Datenträger konnte nicht belegt werden
	X'01'	X'0576'	Widersprüchliche Operandenkombination oder reservierte Felder des Parameterbereiches verwendet oder Selektion enthält große Dateien
	X'20'	X'0577'	Interner Fehler beim Zugriff auf die Auftragsumgebung
	X'82'	X'0594'	Nicht genug virtueller Speicher verfügbar (auch bei einer Auswahlangabe (Wildcard), wenn zu viele Dateien selektiert wurden)
	X'01'	X'0599'	Operand wird in der RFA-BS-Version nicht unterstützt
	X'01'	X'05A8'	Angeforderter Gerätetyp im System nicht gefunden
	X'01'	X'05AB'	Adresse des Ausgabebereiches falsch oder nicht angegeben
	X'82'	X'05B0'	Zurzeit kein passendes Gerät verfügbar
	X'40'	X'05B4'	Datenträger kann nicht bereitgestellt werden
	X'01'	X'05B7'	Fehlerhafter Pfadname angegeben
	X'20'	X'05C7'	Interner Fehler im DMS
	X'40'	X'05D1'	Fehler bei der Geräteanforderung
	X'01'	X'05EA'	VTOC=YES mit teilqualifizierten Dateinamen oder Dateigenerationsgruppen unzulässig
	X'01'	X'05EE'	Dateiname zu lang
	X'40'	X'05FC'	Angegebene Benutzerkennung nicht im Home-Pubset
	X'40'	X'0616'	Ausgabe auf Grund defekter Volume-Sets nicht möglich; falls IFSVSETI = IFSODVSE (ONE DEFECT VOLUMESET), dann steht in IFSVSET die Kennung des Volume-Sets.
	X'01'	X'06B8'	Fehlerhafter Operand angegeben
	X'01'	X'06C7'	Ungültige Generationsnummer angegeben

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'xx' X'01' X'02' X'03'	X'00'	X'06CB'	Ausgabebereich zu kurz Kataloginformation nicht vollständig übertragen Statistische Information nicht vollständig übertragen Kataloginformation und statistische Information nicht vollständig übertragen
	X'40'	X'06CC'	nur bei Auswahlangabe (Wildcard): Keine Datei entspricht der Auswahlangabe
	X'01'	X'06FD'	Parameterbereich ungültig oder nicht zugreifbar
	X'40'	X'06FF'	BCAM-Verbindung abgebrochen
	X'01'	X'FFFF'	Falsche Funktionsnummer im Parameterbereichs-Header
	X'03'	X'FFFF'	Falsche Versionsnummer im Parameterbereichs-Header

Hinweise zur Programmierung (VERSION=4)

Mit FSTAT VERSION=4 werden Dateien auf Net-Storage-Volumes unterstützt.

Der Operand STOTYPE ermöglicht die Auswahl nach dem Speichertyp: Dateien auf gemeinschaftlichen Datenträgern oder Dateien auf Net-Storage-Volumes.

Der Operand FILTYPE ermöglicht die Auswahl nach dem Dateityp: BS2000-Dateien oder Node-Files.

Im Ausgabebereich wird in den Ausgabestrukturen OUTPUT=STAT-LONG/STAT-SHORT/STAT-INFO in den statistischen Informationen von MAIN_HEADER, PVSID_HEADER und USERID_HEADER zusätzlich die Information über die Anzahl von Dateien auf Net-Storage-Volumes ausgegeben (siehe Seite 659):

– Anzahl der Dateien auf Net-Storage (4 Byte)
– Anzahl der freien PAM-Seiten auf Net-Storage (4 Byte)

Im Ausgabebereich werden in der Ausgabestruktur OUTPUT=CEINFO folgende Informationen zusätzlich ausgegeben:

– im Organization-Block (siehe *BLOCK 4 : FILE ORGANIZATION INFORMATION* in Dsect):
  – Dateityp auf Net-Storage (`FILE_TYPE`): BS2000-Datei oder Node-File
  – Dateigröße von Node-Files (`NODE_FILE_SIZE`)
  – Gültigkeitsindikator für die Node-File-Größe (`NODE_FILE_SIZE_VALID`)
– im Allocation-Block (siehe *BLOCK 6A: FILE ALLOCATION INFORMATION* in Dsect):
  – Last Byte Pointer für PAM-Dateien und Node-Files (`LAST_BYTE_POINTER`):
    zeigt auf der letzten logischen Seite der Datei auf das letzte gültige Byte
  – Gültigkeitsindikator für den Last Byte Pointer (`LAST_BYTE_POINTER_VALID`)

Weitere Hinweise enthält der nachfolgende Abschnitt „Hinweise zur Programmierung (VERSION=2, 3 und 4)“.

Hinweise zur Programmierung (VERSION=2, 3 und 4)

Mit FSTAT VERSION=2/3/4 werden die entsprechenden Daten (Extent-Liste und Datenfelder für File-Size und Last Page Pointer) bereits als 4-Byte-Felder zurückgegeben. Diese Schnittstellen brauchen daher für den Aufruf in Konfigurationen mit Dateien > 32 GB nicht umgestellt werden.
Es muss jedoch das Semantikproblem beim Makro OPEN beachtet werden, siehe Seite 753. Jeder Anwender dieser Schnittstelle sollte prüfen, ob dieses Problem auf seine Implementierung zutrifft.

Über den Makroaufruf FSTAT fordert der Anwender Informationen aus dem Katalogeintrag einer oder mehrerer Dateien an.
Zum einen können Informationen aus dem Katalogeintrag ausgegeben werden, für die über den Operanden OUTAREA ein Ausgabebereich reserviert wird. Andererseits existieren statistische Informationen aus dem Katalogeintrag, denen über den Operanden STOUTAR ein Ausgabebereich zugewiesen wird.

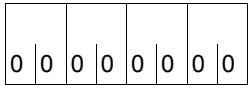
Der zugewiesene Ausgabebereich sollte vor jeder Übertragung mit X'00' überschrieben werden, da nicht benötigter Ausgabebereich nicht bis zum Ende gelöscht wird.

Die Information wird blockweise in den Ausgabebereich übertragen. Die Ausgabe der angeforderten Kataloginformationsblöcke erfolgt direkt hintereinander. Wenn die Anzahl der Dateien bekannt ist, ist eine genaue Berechnung der Größe des Ausgabebereiches möglich. Ist der noch freie Ausgabebereich kleiner als ein noch zu übertragender Informationsblock, wird der Ausgabebereich nicht weiter beschrieben; der noch freie Ausgabebereich wird mit X'00' aufgefüllt. Das Feld IROLN (Real Output Length) des Eingabebereichs gibt Auskunft, wie viele Byte in den Ausgabebereich übertragen worden sind. (Generierung des Eingabeparameterbereichs mit FSTAT MF=D, VERSION=2, XPAND=PLSHORT).

Je nach Anzahl der zu übertragenden Informationsblöcke und der festgesetzten Größe des Ausgabebereiches lassen sich folgende Fälle unterscheiden:

Die Ausgabe war vollständig möglich

Der Ausgabebereich wird mit der gewünschten Information nur so weit wie erforderlich beschrieben, alle unbenutzten Felder enthalten X'00'.
Der Aufrufer erhält den Returncode:



Information vollständig in den Ausgabebereich übertragen

Keine Datei entsprach den Selektionskriterien

Der Ausgabebereich wird nicht beschrieben. Der Aufrufer enthält den Returncode:

0	0	0	0	0	5	3	3

Angegebene Datei in Pubset '(&00)' nicht gefunden

oder

0	0	0	0	0	6	C	C

Keine Datei entspricht den angegebenen Auswahlkriterien

Die Ausgabe war nicht möglich

Der Ausgabereich kann überhaupt nicht beschrieben werden. Der Aufrufer erhält den Returncode:

0	0	0	1	0	5	A	B

Fehlerhafte Bereichsadresse oder falsche Längenangabe im FSTAT-Makro

Das Programm sollte überprüft werden.

Die Ausgabe war nicht vollständig möglich

Der Ausgabebereich ist zu klein bemessen, einige der geforderten Informationsblöcke können nicht übertragen werden. Neben dem Maincode 06CB („Die Längenangabe ist für den Eintrag zu klein“) wird im Subcode 2 hinterlegt, welcher der beiden Ausgabebereiche (OUTAREA/STOUTAR) nicht übertragen werden konnte.

0	1	0	0	0	6	C	B

Katalogeintragsinformationen konnten nicht vollständig übertragen werden (bei OUTPUT=CEINFO/FNAM-ONLY)

0	2	0	0	0	6	C	B

Statistische Informationen konnten nicht vollständig übertragen werden (bei OUTPUT=STAT-SHORT/STAT-LONG)

0	3	0	0	0	6	C	B

Katalogeintrags- und statistische Informationen konnten nicht vollständig übertragen werden (bei OUTPUT=STAT-INFO)

Der Benutzer kann einen größeren Ausgabebereich festlegen und den FSTAT-Aufruf wiederholen.

Inhalt und Struktur der ausgegebenen Information sind von der mit dem Operanden OUTPUT ausgewählten Information abhängig. Es bestehen folgende Möglichkeiten:

OUTPUT = CEINFO / FNAM-ONLY / RC-ONLY / STAT-INFO / STAT-LONG / STAT-SHORT

Für OUTPUT=RC-ONLY wird kein Ausgabebereich benötigt, bei OUTPUT=STAT-INFO werden die Informationen für CEINFO + STAT-LONG ausgegeben.

Auf die Informationsblöcke kann über einen Informationsheader zugegriffen werden. Werden Blöcke nicht angefordert, so wird der Adressverweis dieses Blocks mit Null versorgt.

Die Abstände in den jeweiligen Ausgabebereichen beziehen sich relativ auf den Anfang des Blocks, in welchem die Distanz definiert ist. Die Distanzen im BLOCK_INFORMATION_Header2 beziehen sich auf BLOCK_INFORMATION_Header1, da beide Blöcke als eine Einheit betrachtet werden.

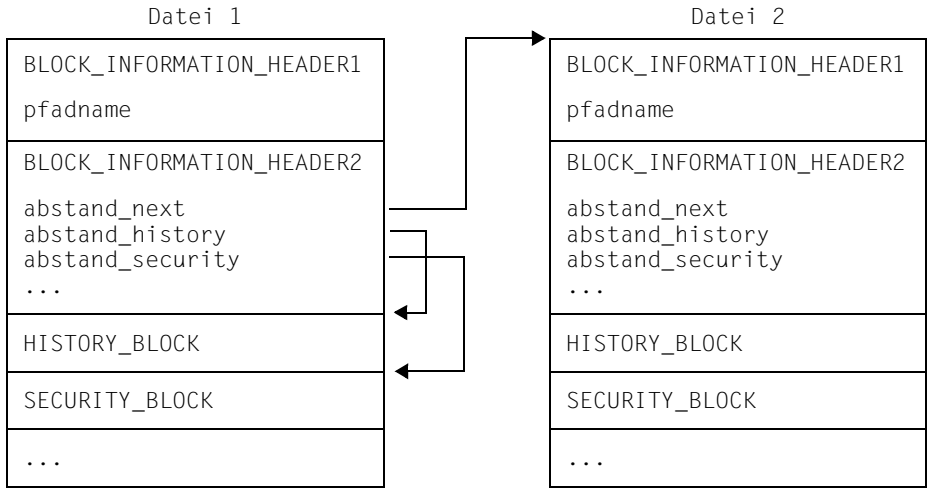
Alle nachfolgend dargestellten Ausgabestrukturen sind beispielhaft und beziehen sich mit ihren Blocknamen (z.B. BLOCK_INFORMATION_HEADER) auf die Blocküberschriften der DSECT, die zur Auswertung des Ausgabebereichs erzeugt werden kann.
Mit dem Makroaufruf **FSTAT MF=D, XPAND=OUTPUT, VERSION=3** wird diese DSECT erzeugt.

Ausgabestruktur: OUTPUT=CEINFO

Bei dieser Ausgabeform werden alle Informationen des Katalogeintrags ausgegeben. Über den Operanden CEINFO können die gewünschten Informationen ausgewählt werden.

CEINFO =	HISTORY	Daten über die Zugriffe auf die Datei
	SECURITY	Daten über die Zugriffsrechte und Datensicherheitsmerkmale
	BACKUP	Daten über Dateisicherung
	ORGANIZATION	Daten über Dateiorganisation
	STATUS	Daten über besondere Eigenschaften der Dateien
	ALLOCATION	Daten über die physikalischen Eigenschaften der Datei
	VOLUME	Datenträgerliste
	VOLUME-EXTENTS	Datenträgerliste und Extent-Liste
	INDEX-INFO	Daten der Dateigenerationsgruppe
	FTAM	Daten der FTAM (File-Transfer-Access-Method)

Es folgt die Darstellung der Ausgabebereiche für zwei Dateien:



Der **BLOCK_INFORMATION_HEADER1** enthält den

pfadname — Pfadname der ausgewählten Datei. Dieser unterteilt sich in:

- Länge der Pvsid (Katalogkennung) (2 Byte)
- Pvsid (Katalogkennung) (4 Byte)
- Länge der Userid (Benutzerkennung) (2 Byte)
- Userid (Benutzerkennung) (8 Byte)
- Länge des Dateinamens (2 Byte)
- Dateiname (41 Byte)

Der **BLOCK_INFORMATION_HEADER2** enthält die Adressverweise auf die Informationsblöcke und den Ausgabebereich der nächsten Datei.

abstand_next — Abstand zum nächsten BLOCK_INFORMATION_HEADER1 (2 Byte) (Beginn des Ausgabebereichs der nächsten Datei)
abstand_history — Abstand zum HISTORY_BLOCK (2 Byte)
abstand_security — (2 Byte)
abstand_backup — (2 Byte)
...

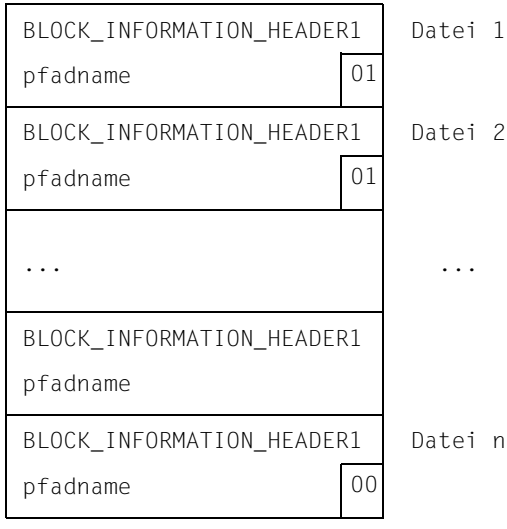
An die beiden Header schließen sich die Informationsblöcke an, welche die eigentliche Nutzinformation enthalten. (HISTORY_BLOCK, SECURITY_BLOCK usw.). Beschreibung und Länge der einzelnen Felder können der DSECT entnommen werden.

Ausgabestruktur: OUTPUT=FNAM-ONLY

Bei dieser Ausgabeform werden nur die Pfadnamen der ausgewählten Dateien in den Bereich OUTAREA ausgegeben. Der BLOCK_INFORMATION_HEADER1 der Ausgabestrukturen OUTPUT=FNAM-ONLY und OUTPUT=CEINFO (siehe Seite 655) entsprechen sich bis auf ein Endekriterium im letzten Byte.

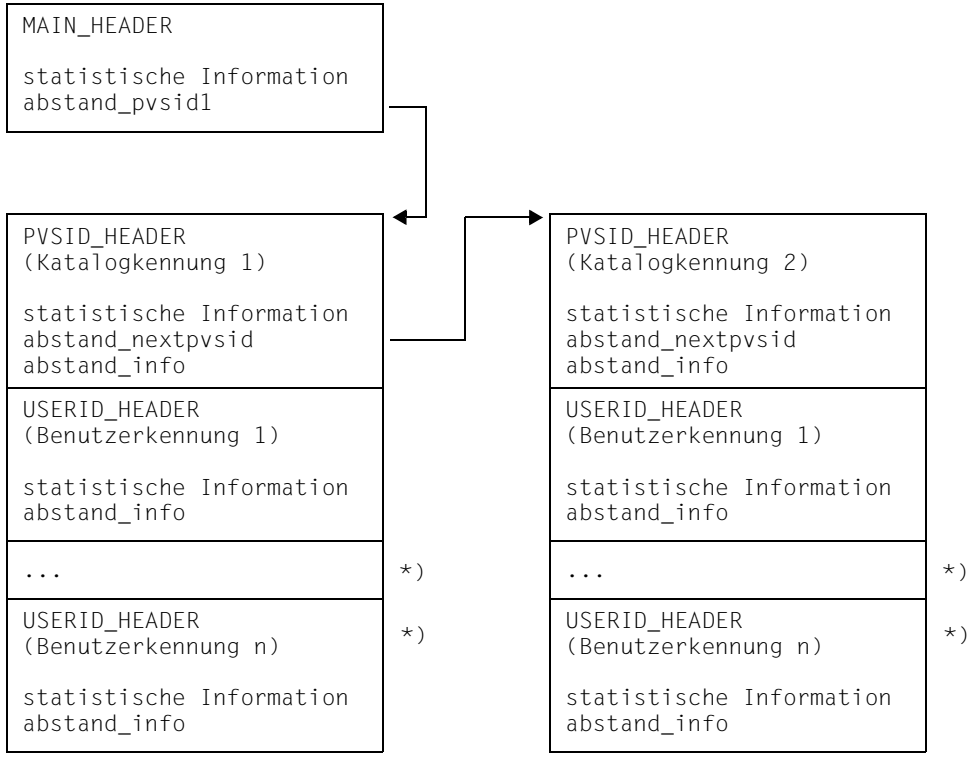
Endekriterium

X'00'	Es folgen keine weiteren Pfadnamen
X'01'	Es folgen weitere Pfadnamen



Ausgabestruktur: OUTPUT=STAT-LONG

Bei dieser Ausgabeform werden nur die statistische Daten in den Ausgabebereich STOUTAR ausgegeben.
Dem *nichtprivilegierten* Benutzer werden Informationen zu Dateien ausgegebenen, die auf einem oder mehreren Pubsets abgelegt sein können.
Der *Systemverwalter* kann sich durch Angabe von Musterzeichen zusätzlich die Informationen zu mehreren Benutzerkennungen ausgeben lassen.



*) Auswahl mehrerer Benutzerkennungen nur durch den *Systemverwalter* möglich.

Der **MAIN_HEADER** liefert folgende Informationen:

Statistische Information

– Gesamtanzahl der ausgewählten Dateien (4 Byte)
– Anzahl der Dateien auf gemeinschaftlichen Datenträgern (4 Byte)
– Anzahl der Dateien auf privaten Datenträgern (4 Byte)
– Anzahl der Dateien auf Net-Storage (4 Byte)
– Anzahl der Dateien auf Magnetband (4 Byte)
– Anzahl der Dateien mit Verdrängungsstufe 1(HSMS) (4 Byte)
– Anzahl der Dateien mit Verdrängungsstufe 2(HSMS) (4 Byte)
– Anzahl der Katalogkennungen (2 Byte)
– Anzahl der freien PAM-Seiten auf gemeinschaftlichen Datenträgern (4 Byte)
– Anzahl der freien PAM-Seiten auf Privatplatte (4 Byte)
– Anzahl der freien PAM-Seiten auf Net-Storage (4 Byte)
– Anzahl der freien PAM-Seiten auf Verdrängungsstufe 1 (4 Byte)
– Anzahl der freien PAM-Seiten auf Verdrängungsstufe 2 (4 Byte)

abstand_pvsid1 Abstand zum ersten PVSID_HEADER (2 Byte)

Der **PVSID_HEADER** liefert folgende Informationen:

Statistische Information

– Katalogkennung (4 Byte)
– Gesamtanzahl der ausgewählten Dateien (4 Byte)
– Anzahl der Dateien auf gemeinschaftlichen Datenträgern (4 Byte)
– Anzahl der Dateien auf privaten Datenträgern (4 Byte)
– Anzahl der Dateien auf Net-Storage (4 Byte)
– Anzahl der Dateien auf Magnetband (4 Byte)
– Anzahl der Dateien mit Verdrängungsstufe 1(HSMS) (4 Byte)
– Anzahl der Dateien mit Verdrängungsstufe 2(HSMS) (4 Byte)
– Anzahl der Benutzerkennungen (2 Byte)
– Anzahl der freien PAM-Seiten auf gemeinschaftlichen Datenträgern (4 Byte)
– Anzahl der freien PAM-Seiten auf Privatplatte (4 Byte)
– Anzahl der freien PAM-Seiten auf Net-Storage (4 Byte)
– Anzahl der freien PAM-Seiten auf Verdrängungsstufe 1 (4 Byte)
– Anzahl der freien PAM-Seiten auf Verdrängungsstufe 2 (4 Byte)

abstand_nextpvsid Abstand zum nächsten PVSID_HEADER (2 Byte)

abstand_info Abstand zum BLOCK_INFORMATION_HEADER1 der ersten ausgewählten Datei dieses Katalogs. (4 Byte) (Abstand relativ zur Adresse des Bereichs OUTAREA)

Der **USERID_HEADER** liefert folgende Informationen:

Statistische Information

- Benutzerkennung (8 Byte)
- Gesamtanzahl der ausgewählten Dateien (4 Byte)
- Anzahl der Dateien auf gemeinschaftlichen Datenträgern (4 Byte)
- Anzahl der Dateien auf privaten Datenträgern (4 Byte)
- Anzahl der Dateien auf Net-Storage (4 Byte)
- Anzahl der Dateien auf Magnetband (4 Byte)
- Anzahl der Dateien mit Verdrängungsstufe 1(HSMS) (4 Byte)
- Anzahl der Dateien mit Verdrängungsstufe 2(HSMS) (4 Byte)
- Anzahl der freien PAM-Seiten auf gemeinschaftlichen Datenträgern (4 Byte)
- Anzahl der freien PAM-Seiten auf Privatplatte (4 Byte)
- Anzahl der freien PAM-Seiten auf Net-Storage (4 Byte)
- Anzahl der freien PAM-Seiten auf Verdrängungsstufe 1 (4 Byte)
- Anzahl der freien PAM-Seiten auf Verdrängungsstufe 2 (4 Byte)

abstand_info — Abstand zum BLOCK_INFORMATION_HEADER1 der ersten ausgewählten Datei dieser Benutzerkennung, die sich unter der oben genannten Katalogkennung befindet (4 Byte) (Abstand relativ zur Adresse des Bereichs OUTAREA)

Ausgabestruktur: OUTPUT=STAT-SHORT

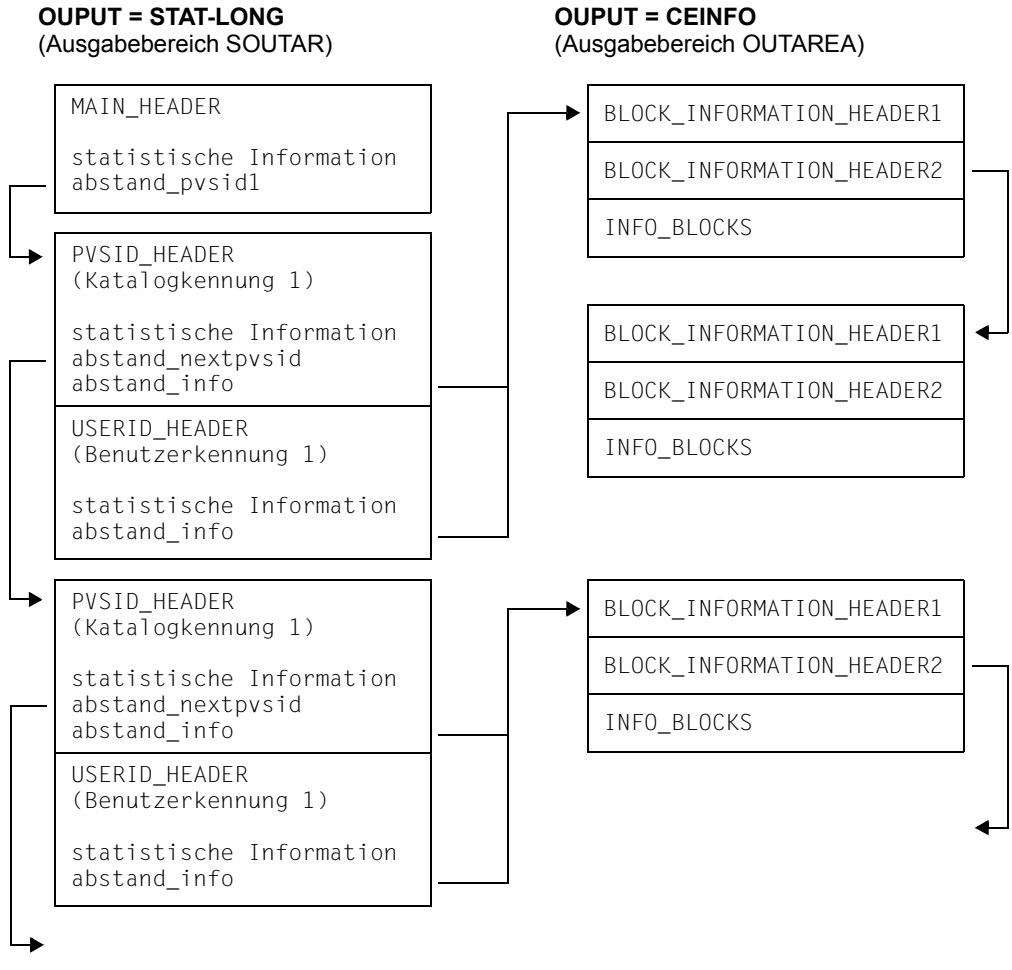
Bei diesem Ausgabeformat wird nur der MAIN_HEADER der Ausgabestruktur OUTPUT= STAT-LONG ausgegeben (siehe Seite 658).

```
MAIN_HEADER

statistische Information
```

Ausgabestruktur: OUTPUT=STAT-INFO

Diese Ausgabe verbindet die Strukturen für OUTPUT=CEINFO und OUTPUT=STAT-LONG. Hierfür muss der Benutzer zwei Ausgabebereiche, OUTAREA und STOUTAR definieren. Der statistische Ausgabebereich STOUTAR (im folgenden Bild links dargestellt) enthält Verweise auf den Ausgabebereich OUTAREA (rechts dargestellt).
Im Ausgabebereich STOUTAR wird nur eine Benutzerkennung (entspricht der Auswahlmöglichkeit des nichtprivilegierten Benutzers) dargestellt.



Hinweise zur Programmierung (VERSION=0 und 1)

Bei Zugriff des FSTAT auf Pubsets mit großen Volumes, auf denen jedoch keine großen Dateien erlaubt sind, verhalten sich die Schnittstellen unverändert. Solche Zugriffe sind also immer problemfrei.
Probleme können entstehen, wenn der Zugriff auf Pubsets erfolgt, auf denen auch große Dateien erlaubt sind.

Schnittstellenvarianten ohne Umstellungsaufwand

```
FSTAT ...,VERSION=0,<teilqualifizierter Dateiname>
FSTAT ...,VERSION=0,<Dateigenerationsgruppe>,GEN=YES
```

```
FSTAT ...,VERSION=1,FNAM
```

Es muss jedoch das Semantikproblem beim Makro OPEN beachtet werden, siehe Seite 753. Jeder Anwender dieser Schnittstelle sollte prüfen, ob dieses Problem auf seine Implementierung zutrifft.

Schnittstellenvarianten mit Prüfungs- / Umstellungsaufwand

```
FSTAT ...,VERSION=0/1,SHORT/LONG
```

Diese Varianten liefern als Ausgabe die Kataloginformation im Format BS2000 V10.0. Die Extent-Liste und die Datenfelder für File-Size und Last Page Pointer werden bei der Ausgabe nur mit 3 Byte dargestellt. Layoutänderungen an diesen Schnittstellen sind aus Kompatibilitätsgründen nicht möglich.

Aufrufe, die große Objekte betreffen, führen zu einem Überlauf der 3-Byte-Datenfelder.

An diesem Punkt werden zwei, von der Treffermenge des FSTAT-Aufrufs abhängige Fälle unterschieden:

a) In der Treffermenge (Menge der selektierten Dateien) existiert keine Datei ≥ 32 GB.

   FSTAT toleriert in diesem Fall den Überlauf des 3-Byte-Datenfeldes der PHP in der Extent-Liste. Die nicht darstellbaren PHPs erhalten den Wert X'FFFFFF' zugewiesen. Es wird dabei davon ausgegangen, dass eine Auswertung der PHPs an den Schnittstellen nie oder äußerst selten erfolgt.

   Damit wird folgendes erreicht:

   – Die Einführung großer Volumes kann kompatibel erfolgen, Anwenderprogramme müssen nicht geändert werden.

   – FSTAT-Aufrufe mit vollqualifizierten Pfadnamen werden kompatibel (bis auf den Überlauf der PHP in der Extent-Liste) unterstützt.

   ► Es ist keine Umstellung nötig.

b) In der Treffermenge existiert mindestens eine Datei ≥ 32 GB (FSTAT wird mit teilqualifiziertem Dateinamen oder Wildcards aufgerufen).

Solche Aufrufe werden mit dem folgenden Returncode zurückgewiesen:

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'01'	X'0576'	Selektion enthält große Dateien

Betroffen sind folgende Typen von FSTAT-Schnittstellen:

Typ		Bemerkung
I	`FSTAT ...`	per Default wird VERSION=0 gesetzt
II	`FSTAT ...,VERSION=0`	
III	`FSTAT ...,VERSION=1,SHORT`	
IV	`FSTAT ...,VERSION=1,LONG`	
V	`FSTAT ...,VERSION=1`	per Default wird der Operand SHORT gesetzt

► Diese Aufrufe müssen auf VERSION=2/3 umgestellt werden.

Übersicht über FSTAT-Aufrufe

```
FSTAT <vollqualifizierter Pfadname>,VERSION=0/1
```

1. nur Zugriff auf kleine Dateien (< 32GB):
   keine Aktion notwendig [1)]
2. Zugriff auch auf große Dateien (≥ 32GB):
   Falls der Aufruf über einen der Typen `I-V` erfolgt, muss auf VERSION=2/3 umgestellt werden, außer bei Typ `I` und `II` mit Angabe einer Dateigenerationsgruppe und GEN=YES.

```
FSTAT <teilqualifizierter Pfadname oder Wildcards im Pfadnamen>,VERSION=0/1
```

1. in der Treffermenge liegen nur kleine Dateien (< 32GB):
   keine Aktion notwendig [1)]

   Achtung
   Prüfen Sie sorgfältig, ob diese Voraussetzung immer erfüllt ist. Dies ist vermutlich nur bei Pfadnamen möglich, die Wildcards in der Katalogkennung haben und deren restliche Bestandteile vollständig (vollqualifiziert) sind.

[1] „keine Aktion notwendig“ gilt hier für den Fall, dass in der Treffermenge nur kleine Dateien existieren und der Überlauf des 3-Byte-Datenfeldes der PHP in der Extent-Liste keine Probleme bereitet. Andernfalls muss auf VERSION=2/3 umgestellt werden!

2. in der Treffermenge können auch große Dateien liegen (≥ 32GB):
   Falls der Aufruf über einen der Typen `III-V` erfolgt, muss auf VERSION=2/3 umgestellt werden.

Zusammenfassung

Bei Verwendung der FSTAT-Schnittstelle mit VERSION < 2 und FORM=LONG oder FORM=SHORT ist eine Umstellung auf die neueste Schnittstellenversion unter folgenden Randbedingungen erforderlich:

a) Im betroffenen Programm soll mit dem FSTAT-Aufruf auf große Dateien zugegriffen werden können.

b) Der Pfadname im FSTAT-Aufruf ist teilqualifiziert oder enthält Wildcards und es kann nicht ausgeschlossen werden, dass in der Treffermenge große Dateien enthalten sind. Besonders kritisch sind hier Aufrufe mit Wildcard in der Katalogkennung.

Dabei müssen neben der Schnittstelle ggf. auch Datenstrukturen im Programm umgestellt werden.

Steuerung über Klasse-2-Systemparameter FST32GB

FST32GB hat nur Einfluss auf folgende FSTAT-Schnittstellen

– Version=0 (entspricht Version=710) bei Angabe eines vollqualifizierten Dateinamens (jedoch nicht bei Angabe einer Dateigenerationsgruppe mit GEN=YES)

– Version=1 (entspricht Version=800), bei der nicht der Operand FNAM spezifiziert wurde

Der Systembetreuer stellt systemglobal ein, ob das Vorhandensein einer Datei ≥ 32 GB in der Menge der selektierten Dateien zur Abweisung des FSTAT-Aufrufs mit dem Returncode `X'00000576'` führt (FST32GB=0, Standardeinstellung) oder ein Überlauf der 3-Byte-Datenfelder generell toleriert wird (FST32GB=1). Im letzten Fall wird den nicht darstellbaren Datenfeldern der Wert X'FFFFFF' zugewiesen.

Hinweis

Der Klasse-2-Systemparameter FST32GB wird nicht ausgewertet, wenn der FSTAT-Indikator (siehe unten) gesetzt ist.

Steuerung über FSTAT-Indikator

Ein Verhalten wie bei FST32GB=1, also das Ignorieren des Überlaufs, kann schnittstellenspezifisch bei den FSTAT-Typen `I-V` über einen Indikator aktiviert werden.

Hinweis

Der FSTAT-Indikator besitzt eine höhere Priorität als der Klasse-2-Systemparameter FST32GB. Wird der FSTAT-Indikator gesetzt, wird FST32GB nicht ausgewertet.
Der Indikator muss direkt in der Parameterliste gesetzt werden; eine Unterstützung im FSTAT-Makro ist nicht vorhanden.

Beschreibung der Bits in den entsprechenden DSECTs:

```
         FSTAT MF=D,PARMOD=31,VERSION=710
IDBFLAG2 DS    X                              FLAGS 2
IDBLOPYE EQU   X'04'                          2-2 S LARGE PUBSET ACCESS=YES

         FSTAT MF=D,PARMOD=31,VERSION=800
IFLAG0   DC    B'10001100'
ILOPY    EQU   X'04'                          2-2 S LARGE PUBSET ACCESS=YES
```

Versionsunterschiede – VERSION=0/1/2/3/4

Die folgende Tabelle zeigt, welche Operanden/Operandenwerte mit VERSION=4/3/2/1/0 unterstützt werden.

– Mit VERSION=4 können alle Operanden/Operandenwerte verwendet werden, die ab der Version BS2000/OSD-BC V9.0A unterstützt werden.

– Mit VERSION=3 können alle Operanden/Operandenwerte verwendet werden, die bis einschließlich der Version BS2000/OSD-BC V8.0A unterstützt wurden.

– Mit VERSION=2 können alle Operanden/Operandenwerte verwendet werden, die bis einschließlich der Version BS2000/OSD-BC V2.0A unterstützt wurden.

– Mit VERSION=1 können alle Operanden/Operandenwerte verwendet werden, die in den BS2000-Versionen V8.0A bis V10.0A unterstützt wurden.

– Mit VERSION=0 können alle Operanden/Operandenwerte verwendet werden, die in den BS2000-Versionen < V8.0A unterstützt wurden.

Operand	V=0	V=1	V=2	V=3	V=4	Bemerkungen zu Operandenwerten
MF=E	x	x	x	x		
PARMOD	x	x	x	x		
VERSION	x	x	x	x		
MF=C	-	-	x	x		
MF=D	x	x	x	x		
PARMOD	x	x	x	x		
PREFIX	-	x	x	x		
VERSION	x	x	x	x		
MF=L	x	x	x	x		
pfadname	x	x	x	x		Vers=0: Temporäre Dateien werden nicht berücksichtigt, siehe auch [3)] Vers=1: Die Länge des Pfadnamens wird über einen Stellungsoperanden angegeben, siehe [3)]
ACCESS	-	x	x	x		ANY erst ab Vers=2
ACL	-	x	x	x		ANY erst ab Vers=2
ACCCNT	-	-	x	x		
ADMINFO	-	-	-	x		
AVAIL	-	-	-	x		
BACKUP	-	x	x	x		ANY erst ab Vers=2
BASACL	-	x	x	x		ANY erst ab Vers=2; YES mit Unterstruktur bei Vers=1, siehe [2)].

Operand	V=0	V=1	V=2	V=3	V=4	Bemerkungen zu Operandenwerten
MF=L (Fortsetzung)						
BLKCNT	-	-	x	x	x	
BLKCTRL	-	x	x	x	x	Vers=1: Nur NONE, DATA, PAMKEY, NO sind möglich (auch als Liste anzugeben)
CEINFO	-	-	x	x	x	
CCS	-	-	x	x	x	
CRDATE	-	x	x	x	x	ANY, NONE und Werte mit der Angabe (zeit) erst ab Vers=2, siehe auch [1)]
DELDATE	-	-	-	x	x	
DISKWR	-	-	x	x	x	
EXDATE	-	x	x	x	x	ANY, NONE und Werte mit der Angabe (zeit) erst ab Vers=2, siehe auch [1)]
EXTENTS	-	x	x	x	x	ANY erst ab Vers=2
FCBTYPE	-	x	x	x	x	ANY erst ab Vers=2
FSIZE	-	x	x	x	x	ANY erst ab Vers=2
FILTYPE	-	-	-	-	x	
FROM	-	x	x	x	x	
GEN	x	x	x	x	x	
GROUPAR	-	(x)	x	x	x	GROUPAR erst ab Vers=2, Vers=1: Information mit BASACL abrufbar, siehe [2)]
GUARDS	-	-	x	x	x	
IOPREF	-	-	x	x	x	
IOUSAGE	-	-	x	x	x	
LADATE	-	x	x	x	x	ANY, NONE und Werte mit der Angabe (zeit) erst ab Vers=2, siehe auch [1)]
LASTPAG	-	x	x	x	x	ANY erst ab Vers=2
LCDATE	-	-	x	x	x	
MANCLAS	-	-	-	x	x	
MIGRATE	-	x	x	x	x	ANY erst ab Vers=2 FORBIDDEN erst ab Vers=3
OTHERAR	-	(x)	x	x	x	Vers=1: OTHERAR nicht unterstützt, Information mit BASACL abrufbar, siehe [2)]
OUTAREA	(x)	(x)	x	x	x	Vers=0/1: Länge und Adresse des Ausgabebereichs werden mit Stellungsoperanden dargestellt (bei Vers=1: siehe [3)]); *REQUEST und *RELEASE erst ab Vers=3

Operand	V=0	V=1	V=2	V=3	V=4	Bemerkungen zu Operandenwerten
MF=L (Fortsetzung)						
OUTPUT	(x)	(x)	x	x	x	Vers=0: OUTPUT wird nicht unterstützt. Die Stellungs-operanden SHORT und LONG entsprechen den Operandenwerten STAT-SHORT und STAT-LONG des Schlüsselwortoperanden OUTPUT. Siehe auch [3)] Vers=1: OUTPUT wird nicht unterstützt. Die Stellungsoperanden FNAM, SHORT und LONG entsprechen den Operandenwerten FNAM-ONLY und CEINFO des Schlüsselwortoperanden OUTPUT. Siehe auch [3)]
OWNERAR	-	(x)	x	x	x	Vers=1: OWNERAR wird nicht unterstützt, Information mit BASACL abrufbar, siehe [2)]
PASS	-	x	x	x	x	ANY erst ab Vers=2
PASSW	-	-	x	x	x	
PREFIX	-	x	x	x	x	
PREFORM	-	-	-	x	x	
PROTACT	-	-	x	x	x	
RELSPAC	-	-	x	x	x	
SHARE	-	x	x	x	x	ANY und Angabe einer Liste erst ab Vers=2
SIZE	-	x	x	x	x	ANY erst ab Vers=2
SLEVEL	-	x	x	x	x	ANY erst ab Vers=2
SORT	-	x	x	x	x	
STATE	x	x	x	x	x	Vers=0: Nur NOCLOS ist möglich Vers=1: Nur NOCLOSE und PCLOSE sind möglich Vers=2: CACHE-NOT-SAVED und DEFECT-REPORTED nicht möglich
STOCLAS	-	-	-	x	x	
STOTYPE	-	-	-	-	x	
STOUTAR	(x)	(x)	x	x	x	Vers=0/1: Länge und Adresse des Ausgabebereichs werden mit Stellungsoperanden dargestellt. In beiden Versionen wird nur ein Ausgabebereich definiert; die Ausgabeinformation wird mit den Stellungsoperanden SHORT, LONG und FNAM (nur Vers=1) festgelegt. Siehe [3)]
SUPPORT	x	x	x	x	x	Angabe einer Liste erst ab Vers=1 ANY erst ab Vers=2
S0MIGR	-	-	-	x	x	

Operand	V=0	V=1	V=2	V=3	V=4	Bemerkungen zu Operandenwerten
TIMBASE	-	-	-	x	x	
TYPE	x	x	x	x	x	Vers=0/1: Nur FGG möglich
USRINFO	-	-	-	x	x	
VERSION	x	x	x	x	x	
VOLSET	-	-	-	x	x	
VOLUME	x	x	x	x	x	ANY erst ab Vers=2
VTOC	x	x	x	x	x	
WORKFIL	-	-	-	x	x	
WTQUIET	-	-	-	x	x	
XPAND	-	-	x	x	x	

Legende

x Operand ist in der Makroversion verfügbar

(x) Operand ist unter dem angebenen Namen in der Makroversion nicht verfügbar; die Funktion ist jedoch mit einem anders lautenden Operanden ausführbar.

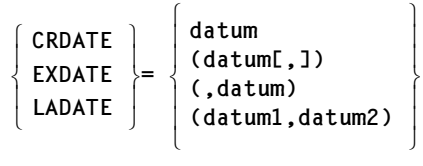
- Operand ist in der Makroversion nicht verfügbar

Vers Version: VERSION=710 entspricht VERSION=0
VERSION=800 entspricht VERSION=1

Hinweis

Der Stellungsoperand pfadname ist vor den alphabetisch sortierten Schlüsselwortoperanden eingeordnet.

1) Die Operanden CRDATE, EXDATE und LADATE haben bei Vers=1 folgendes Format:



2) Der Operand BASACL hat bei Vers=1 folgendes Format:

```
                  ┌ NONE                                                  ┐
                  │ YES([,OWNER={ NO-ACCESS      }]                       │
                  │            { zugriffsliste   }                       │
[,BASACL={        │ [,GROUP={ NO-ACCESS      }]                          }]
                  │         { zugriffsliste  }                           │
                  │ [,OTHERS={ NO-ACCESS     }])                         │
                  │          { zugriffsliste }                           │
                  └                                                       ┘
```

3) Darstellung der Stellungsoperanden bei Vers=0 und Vers=1:

```
Vers=0: [pfadname],ausdr[,länge][,{ SHORT }]
                                  { LONG  }

                { [pfadname]           }  { ausadr     }  { länge2 }  [,{ SHORT }]
Vers=1:{        { ([pfadname],länge1)  }, { (S,ausadr) }, { (r2)   }   { FNAM  }]
                                          { (r1)       }              { LONG  }
```

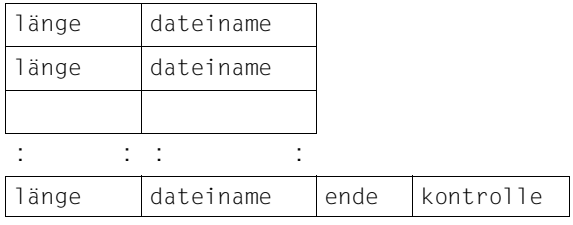
Versionsunterschiede bei der Darstellung des Ausgabebereiches

Version=0 (≙ Version=710)

Der Umfang der Informationen, die in den Ausgabebereich geschrieben werden, ist davon abhängig, ob ein teilqualifizierter Dateiname, ein vollqualifizierter Dateiname einer Dateigenerationsgruppe oder ein vollqualifizierter Dateiname angegeben wird.

Teilqualifizierter Dateiname oder vollqualifizierter Dateiname einer FGG

Es wird nur eine Liste der Dateinamen in den Ausgabebereich geschrieben.



länge — Länge des Dateinamens. Das Längenfeld hat den Wert „Länge des Dateinamens + 1 Byte für die Größe des Längenfeldes“. Hiermit wird der Abstand bis zum nächsten Längenfeld festgelegt.
Dem Namen eines Gruppeneintrages folgt die Zeichenkette „_(FGG)“. Das Längenfeld hat dann den Wert „Länge des Gruppennamens + 1 Byte für die Größe des Längenfeldes + 6 Byte für die Zeichenkette „_(FGG)“.

dateiname — Dateiname (maximal 41 Byte)

ende — zeigt das Listenende an (X'00') (1 Byte).

kontrolle — zeigt an, ob der Ausgabebereich zur Aufnahme aller Dateinamen groß genug war (1 Byte).

- X'00' Ausgabe vollständig.
  Alle Dateinamen sind in den Ausgabebereich übertragen worden.
- X'01' Ausgabe unvollständig
  Einer oder mehrere Dateinamen konnten nicht in den bereits gefüllten Ausgabebereich übertragen werden.

Vollqualifizierte Dateinamen

Das Ausgabeformat wird über die Operanden SHORT und LONG festgelegt.

SHORT Der statistische Teil des Katalogeintrags wird in den Ausgabebereich (mindestens 60 Byte) übertragen. Der Makro IDCE generiert eine Dsect, die das Layout des Ausgabebereichs erzeugt.

LONG Der vollständige Katalogeintrag wird in den Ausgabebereich (mind. 2032 Byte) ausgegeben. Er besteht aus:
- statistischer Teil
- Dateiname
- Erweiterung
- Datenträger-Tabelle
- Datei-Tabelle
- FGG-Zusatz

Wenn die im Operanden „länge“ (in Vers=2 entspricht dieser Operand dem Operandenwert „länge“ der Operanden OUTAREA oder STOUTAR) angegebene Länge des Ausgabebereiches nicht ausreicht, wird keine Information übertragen. (Zur Dsect-Erzeugung für den Ausgabebereich siehe Beschreibung des Operanden SHORT auf Seite 674).

Beispiel

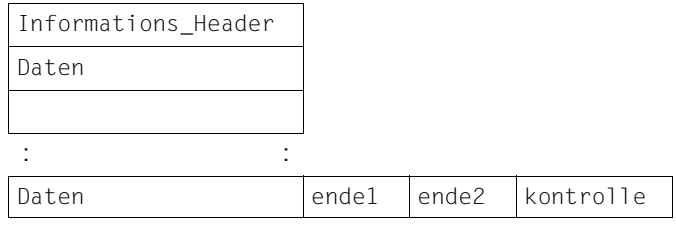
06 KLAUS	0B ABRECHNUNG	00	00

Version=1 (≙ Version=800)

Es werden Informationen von permanenten oder temporären Dateien, von Dateigenerationen oder Dateigenerationsgruppen ausgegeben, die durch Angabe eines voll- oder teilqualifizierten Namens ausgewählt werden.

Größe und Aufbau des Ausgabebereiches hängt vom Informationsumfang ab, der über die Operanden FNAM, SHORT und LONG für die Datei(en) angefordert wird.

FNAM In den Ausgabebereich wird eine Liste der Dateinamen übertragen. Dem Ausgabebereich wird über den Operanden „ausadr“ eine Adresse und über den Operanden „länge2“ die Länge zugewiesen.
(Bei Vers=2 entsprechen diese Operanden den Operandenwerten adr und länge des Operanden OUTAREA.)
Der Ausgabebereich muss mindestens 73 Byte lang sein.



Der Informations_Header enthält:

- Gesamtanzahl der ausgewählten Dateien (4 Byte)
- Länge des folgenden Präfix und der folgenden Daten (4 Byte)
- Präfix:
  - Länge des Präfix (2 Byte)
  - Länge der Katalogkennung + 2 (2 Byte)
  - Katalogkennung für alle folgenden Einträge
  - Länge der Benutzerkennung + 2 (2 Byte)
  - Benutzerkennung für alle folgenden Einträge

Der Bereich Daten enthält:

- Länge der nachfolgenden Nutzinformation + 2 (2 Byte)
- Dateiname (max. 41 Byte)

Der Ausgabebereich wird abgeschlossen mit:

ende1 Listenende-Kennzeichen für den Datenbereich (2 Byte, X'0000')

ende2 Listenende-Kennzeichen für den Ausgabebereich (4 Byte, X'00000000')

kontrolle Rückkehr-Information über die Ausführung des FSTAT-Aufrufes. (1 Byte)

X'00' alle geforderten Informationen wurden in den Ausgabebereich übertragen

X'01' die geforderte Information wurde nur unvollständig in den Ausgabebereich übertragen

SHORT Zu jedem Dateinamen wird die statistische Information des Katalogeintrags ausgegeben.
Dem Ausgabebereich wird über den Operanden „ausadr“ eine Adresse und über den Operanden „länge2“ die Länge zugewiesen.
(Bei Vers=2 entsprechen diese Operanden den Operandenwerten adr und länge des Operanden STOUTAR)
Der Informations_Header und die Bereiche ende1, ende2 und kontrolle entsprechen der Darstellung beim Operanden FNAM.

Der Bereich Daten enthält:
– statistischen Teil des Katalogeintrags. Das letzte Byte enthält die Länge des nachfolgenden Dateinamens (60 Byte).
– Dateiname (max. 41 Byte)

Der Datenbereich ist folglich für SHORT maximal 101 Byte lang. Der Ausgabebereich muss für SHORT mindestens 133 Byte lang sein. Der Makro IDCE generiert eine Dsect, die das Layout des Ausgabebereichs erzeugt.

LONG Zu jedem Dateinamen wird der vollständige Katalogeintrag ausgegeben. Dem Ausgabebereich wird über den Operanden „ausadr“ eine Adresse und über den Operanden „länge2“ die Länge zugewiesen.
(Bei Vers=2 entsprechen diese Operanden den Operandenwerten adr und länge der Operanden OUTAREA und STOUTAR)
Der Informations-Header und die Bereiche ende1, ende2 und kontrolle entsprechen der Darstellung beim Operandenem FNAM.

Der Bereich Daten enthält:
- statistischer Teil
- Dateiname
- Erweiterung
- Datenträger-Tabelle
- Datei-Tabelle
- FGG-Zusatz

Der Ausgabebereich muss für LONG mindestens 2064 Byte lang sein.

Hinweis

Für r_1, r_2 und r_3 dürfen nicht die Mehrzweckregister 1 und 15 angegeben werden. Die Angaben für r_1, r_2 und r_3 müssen sich paarweise ausschließen (entweder-oder-Logik).

GET – Nächsten Satz lesen

ISAM:	Makrotyp:	R bei PARMOD=24 0 bei PARMOD=31
SAM:	Makrotyp:	R bei PARMOD=24 R bei PARMOD=31

Anwendungsbereich

Die Funktion GET ist bei der Dateiverarbeitung mit SAM oder ISAM (satzorientierte Zugriffsmethoden) möglich. Die Datei muss dabei mit einem der folgenden OPEN-Modi eröffnet worden sein:

– INPUT (SAM oder ISAM)
– INOUT (ISAM)
– OUTIN (ISAM)
– UPDATE (SAM)

Funktion

Durch die Funktion GET wird dem Aufrufer ein Satz der Datei zur Verfügung gestellt. Es wird dabei immer der Satz bereitgestellt, den der aktuelle Satzzähler referenziert. Nach einem OPEN INPUT ist dieser Zähler 1. Eine Folge von GET-Aufrufen liest somit die Sätze der Datei sequenziell (vgl. GETKY, GETFL, GETR bei ISAM).

Ein GET-Aufruf setzt nicht jedes Mal einen SVC ab. Ein SVC wird nur dann abgesetzt, wenn der nächste zu lesende Satz nicht mehr im aktuellen Pufferbereich steht.

Wird ein Satz angefordert, der außerhalb der Datei liegt, erkennt das DVS „Dateiende“. Es verzweigt zur EOFADDR-Adresse (siehe Makro EXLST, Seite 394) und übergibt dem Anwender die Steuerung.

Bereitstellungsmodi

Der Aufrufer kann eine Datei in zwei Modi verarbeiten, die für GET wesentlich sind:

Im **LOCATE-Modus** (Ortungsmodus) wird vom System der Satz in einen Pufferbereich des Systems übertragen (Operand IOREG). Der Aufrufer erhält in einem von ihm spezifizierten Register die Adresse des Satzanfangs. Eine Übertragung in den Programmbereich findet nicht statt (vgl. MOVE-Modus).

Im **MOVE-Modus** stellt der Aufrufer in seinem Programm einen Bereich zur Verfügung, in den das System den Satz kopieren kann. Die Adresse dieses Satzbereiches wird über Register 1 dem System beim Aufruf übergeben.

Verändern des Satzzeigers

Der Satzzeiger kann durch Positionierung verändert werden. Dadurch wird die sequenzielle Verarbeitung unterbrochen und an anderer Stelle wieder fortgesetzt : vgl. SETL, SETLKY.

*Besonderheiten nur für **SAM**-Dateien*

OPEN UPDATE
Im OPEN-Modus UPDATE werden mit GET die zu aktualisierenden Sätze gelesen. PUTX verändert den Satz, führt jedoch nicht zu einem Schreibauftrag. Es wird vielmehr nur ein Kennzeichen im FCB gesetzt, dass dieser Satz noch geschrieben wrden muss. Der eigentliche Schreibauftrag erfolgt erst mit einem folgenden GET, RELSE,SETL; PUTX setzt keinen SVC ab.

Banddatei und Blockung
Bei Banddateien im Format PAMKEY und hoher Blockung (STD,n) kommt es zu einer Verlängerung der I/O. Es ist hier ratsam, auf NON-KEY-Format oder BLKSIZE=länge umzusteigen.

Die Zugriffszeit kann gering gehalten werden, wenn große Pufferbereiche benutzt werden. SAM kettet dann seinerseits aufeinander folgende PAM-Seiten (gekettete Ein-/ Ausgabe). Ein merklicher Zeitgewinn ist jedoch nur für solche Dateien zu erwarten, die weitgehend zusammenhängenden Speicherplatz haben (siehe SPACE-Operand im FILE-Makro, Seite 497).

Ein GET-Makroaufruf führt nur dann zu einem SVC, wenn eine Pufferübertragung ausgelöst wird. Der Anwender kann daher nicht erwarten, bei jedem GET-Aufruf die Kontrolle in einem STXIT-Prozess zu bekommen (bei Verwendung des STXIT-Makroaufrufs mit SVC oder SVCLIST. Näheres siehe Handbuch „Makroaufrufe an den Ablaufteil“ [2]).

Die PUTX Verarbeitung „merkt“ sich im TU-FCB, dass im aktuellen Puffer ein Satz aktualisiert worden ist. Wenn beim nächsten GET (RELSE, SETL) dieses Bit gesetzt ist, dann erst wird der ganze logische Block auf Platte geschrieben. Der PUTX setzt **nie** einen SVC ab.

*Besonderheiten nur für **ISAM**-Dateien*

Lesen über Primärschlüssel
Bei ISAM-Dateien wird die logische Folge der Sätze durch Primärschlüssel definiert.
Mit einer Folge von GET-Aufrufen werden die Sätze sequenziell im Sinne aufsteigender Primärschlüssel gelesen.
Durch einen SETLKEY kann auf einen bestimmten Primärschlüssel positioniert werden Ein folgender GET liefert dann diesen Satz.

Lesen über Sekundärschlüssel
Bei der Verarbeitung einer NK-ISAM-Datei kann auch über die Angabe eines Sekundärschlüssels gelesen werden. Existieren zu einem Primärschlüssel mehrere Sätze mit identischen Sekundärschlüsseln (DUPEKEY), so werden diese Sätze durch GET-Aufrufe in der zeitlichen Reihenfolge ihres Entstehens geliefert.

Enthält die Datei Sätze mit gleichen Primärschlüsseln, werden sie in der Reihenfolge geliefert, in der sie in die Datei eingefügt wurden.

Wenn die Datei sequenziell über einen Sekundärschlüssel gelesen wird, so werden Sätze mit gleichen Sekundärschlüsselwerten in der Reihenfolge bereitgestellt, in der die Sekundärschlüsselwerte entstanden sind.

Wird mit SETL KEY auf einen existierenden Satz positioniert, stellt ein nachfolgender GET diesen Satz zur Verfügung.

Format

Operation	Operanden
GET	{fcbadr / (1)}[,{area / (0)}][,{LOCK / NOLOCK}] [,AIX={NO / YES,{KEYNAME=name / KEYNMAD=adr}}] [,PARMOD={24 / 31}]

Operandenbeschreibung

fcbadr
Adresse des FCB der zu verarbeitenden Datei.

Nur für ISAM-Dateien:
Wenn die Datei über einen Sekundärschlüssel gelesen werden soll, muss dieser FCB in seiner 31-Bit-Schnittstelle vorliegen.

(1)
Die FCB-Adresse steht im Register 1.

area
Adresse des Feldes, in das der Satz gebracht werden soll, wenn die Verarbeitung im Move-Mode erfolgt. Im Locate-Mode wird „area“ ignoriert.

(0)
Die Adresse des Feldes, in das der Satz gebracht werden soll, steht im Register 0.

LOCK
Nur für ISAM-Dateien:
Die Satz- oder Datenblocksperre bleibt nach Ausführung des Makroaufrufs erhalten (explizite Sperre).

NOLOCK
Es wird keine explizite Sperre gesetzt.

AIX
Nur für ISAM-Dateien:
Gibt an, ob der Satz über seinen Primär- oder einen Sekundärschlüssel bereitgestellt werden soll.

= NO
Der Satz wird über seinen Primärschlüssel bereitgestellt.

= YES
Kann nur angegeben werden, wenn
- *die 31-Bit-Schnittstelle des Makros generiert wird (über den Operanden PARMOD=31 oder den Makroaufruf GPARMOD 31) und*
- *der Makro sich auf einen 31-Bit-FCB bezieht.*

Der Satz wird über den im Operanden KEYNAME oder KEYNMAD vereinbarten Sekundärschlüssel bereitgestellt.

KEYNAME = name

Nur für ISAM-Dateien:

Gibt den Namen des Sekundärschlüssels an, über den der Satz gelesen werden soll. name muss der Name eines für die aktuelle Datei vereinbarten Sekundärschlüssels sein. Die Namen aller für eine Datei definierten Sekundärschlüssel können mit dem Makro SHOWAIX oder dem Kommando SHOW-INDEX-ATTRIBUTES ermittelt werden.

KEYNMAD = adr

Nur für ISAM-Dateien:

Gibt die symbolische Adresse (den Namen) eines Feldes an, in dem der Anwender den Namen des Sekundärschlüssels hinterlegt hat, über den der Satz gelesen werden soll. Das Feld mit der symbolischen Adresse adr muss zum Zeitpunkt der Makroausführung den Namen eines für die aktuelle Datei vereinbarten Sekundärschlüssels enthalten.

PARMOD

Gibt den Generierungsmodus an.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24

Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (nur 24-Bit-Adressierung).

= 31

Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist (24-Bit- oder 31-Bit-Adressierung).

Hinweis zur Programmierung

Der GET-Makroaufruf zerstört die Register 0, 1, 14 und 15.

GETFL – Satz nach Markierung lesen

Makrotyp: R bei PARMOD=24
S bei PARMOD=24 und Angabe des MF-Operanden
0 bei PARMOD=31

Der GETFL-Makroaufruf kann nur auf Dateien angewendet werden, die mit Markierungen erstellt wurden: er wertet die Markierungen (= Flags) im ISAM-Index aus und stellt dem Anwenderprogramm aus einem bestimmten Bereich der Datei (siehe Operand LIMIT, Seite 684) den nächsten Satz zur Verfügung, der den im GETFL-Aufruf vorgegebenen Bedingungen genügt. Der GETFL-Makro kann sowohl Wertmarkierung als auch logische Markierung auswerten, die Suche kann sowohl in Richtung Dateianfang wie Dateiende verlaufen.

Zu beachten ist, dass NK-ISAM keine Markierungen in die Indexeinträge aufnimmt. Eine Suche nach Sätzen mit bestimmten Merkmalen verläuft daher als sequenzielle Leseoperation.

Für K-ISAM-Dateien werden die Markierungen entsprechend der VALPROP-Angabe in FILE/FCB ausgewertet und in den Indexeintrag übernommen; eine Suche mit GETFL verläuft also nicht sequenziell, sondern über den Indexbaum. Der einzige für den Anwender sichtbare Unterschied in der Flagverarbeitung von NK-ISAM- und K-ISAM-Dateien ist die geringere Performance bei NK-ISAM. In NK-ISAM-Dateien sollte auf Flagverarbeitung verzichtet werden.

Werden VALTEST und LOGTEST angegeben, so müssen auf den gesuchten Satz beide Bedingungen zutreffen.

Werden weder VALTEST noch LOGTEST angegeben, wirkt der GETFL-Makroaufruf (innerhalb der mit LIMIT gesetzten Grenzen) wie ein GET- oder GETR-Makroaufruf. Wird die mit LIMIT definierte Grenze erreicht, geht die Steuerung an den EXLST-Ausgang EOFADDR (bei LIMIT=END) oder an den EXLST-Ausgang NOFIND (bei LIMIT=KEY).

Wird der GETFL-Makroaufruf auf eine Datei angewendet, die ohne Markierungen erstellt wurde, geht die Steuerung an den EXLST-Ausgang USERERR.

Das Feld, auf das der FCB-Operand KEYARG verweist, muss groß genug sein, um den gesamten Index aufnehmen zu können (Schlüssel + Wertmarkierung + logische Markierung). Vorgaben für Wertmarkierungen oder Masken für logische Markierungen müssen formal mit den entsprechenden Markierungen in den Datensätzen übereinstimmen (vor allem hinsichtlich Lage und Länge).

Wertmarkierung

Bei der Suche nach einem Satz mit bestimmter Wertmarkierung (Operand VALTEST) wird der entsprechende Markierungsbereich eines jeden Satzes bzw. im Indexeintrag mit dem beim GETFL-Makroaufruf angegebenen Wert verglichen. Es wird jeweils der Satz gelesen, der innerhalb des Bereichs als erster den Bedingungen genügt.

logische Markierung

Bei der Suche nach einem Satz anhand der logischen Markierung muss im GETFL-Makroaufruf eine Bit-Maske definiert werden, die bitweise mit der logischen Markierung verglichen wird. Entsprechend der Angabe im Operanden LOGTEST wird der erste Satz gelesen, der innerhalb des angegebenen Bereichs als erster einer oder allen Bedingungen genügt.

Formate

Die Formate des GETFL-Makros unterscheiden sich durch den MF-Operanden:

kein MF-Operand	der Makro generiert Parameter und Systemaufruf, PARMOD=24/31, FCB-Adresse als symbolische Adresse oder in Register 1
MF=L	List-Form: der Makro generiert die Operandenliste, PARMOD=24, FCB-Adresse nicht in Register 1
MF=E	Execute-Form: der Makro generiert den Systemaufruf zu MF=L

Da die Parameter im Format ohne MF-Operanden und im Format mit MF=L bis auf FCB-Adresse, PARMOD und MF identisch sind, werden diese beiden Formate nicht getrennt dargestellt.

Operation	Operanden
GETFL	{fcbadr / (1) / (r2)}[,{area / (0)}][,{LOCK / NOLOCK}] [,LIMIT={END / KEY}][,LOGTEST={ANY / ALL}] [,REVERSE=YES] [,VALTEST={GT / GE / EQ / NE / LE / LT}][,PARMOD={24 / 31}]
	[MF={(E,{liste / (r1)}) / L}]

Operandenbeschreibung

fcbadr
Adresse des FCB für die Datei

(1)
Die FCB-Adresse steht im Register 1.

(r2)
nur bei PARMOD=24: bei Ausführung des GETFL mit MF=(E,) steht die FCB-Adresse in dem mit „r2“ bezeichneten Register (r2≠1)

area
Adresse des Bereichs, in den der Satz gebracht werden soll

(0)
Die Adresse des Bereichs, in den der Satz gebracht werden soll, steht im Register 0.

LOCK
Die Sperre bleibt nach Ausführung des Makroaufrufs bestehen (explizite Sperre).

NOLOCK
Die Sperre bleibt nach dem Lesen nicht bestehen.

LIMIT
definiert die obere Grenze des zu durchsuchenden Bereichs, die untere Grenze ist durch die aktuelle Zeigerposition in der Datei gegeben – abhängig vom vorausgegangenen Makroaufruf. Die „Suchrichtung“ hängt davon ab, ob REVERSE=YES gilt. Die Suche beginnt:

- nach einem Makroaufruf SETL B mit dem ersten Satz der Datei
- nach einem Makroaufruf SETL E mit dem letzten Satz der Datei (sinnvoll nur zusammen mit REVERSE=YES)
- nach einem Makroaufruf SETL KEY an der Zeigerposition
- nach anderen Makroaufrufen mit dem ersten Satz vor oder hinter der Zeigerposition (abh. von REVERSE=YES)

= END
Die Suche wird solange fortgesetzt, bis Dateiende (oder Dateianfang) erreicht ist. Enthält die Datei keinen Satz, dessen Markierungen die vorgegebenen Bedingungen erfüllen, wird der EXLST-Ausgang EOFADDR aktiviert.

= KEY
Die Grenze des Bereichs ist durch einen Schlüssel definiert, auf den der FCB-Operand KEYARG verweist. Die Suche wird abgebrochen, wenn der zu prüfende Satz den gleichen Schlüssel hat wie der durch KEYARG referenzierte.

– ohne REVERSE=YES: es werden nur Sätze geprüft, deren Schlüssel kleiner ist als der über KEYARG adressierte Schlüssel.
– mit REVERSE=YES: es werden nur Sätze geprüft, deren Schlüssel größer ist als der über KEYARG adressierte Schlüssel.

Enthält der über einen Schlüssel abgegrenzte Bereich keinen Satz, dessen Markierungen den Bedingungen im GETFL-Makroaufruf entsprechen, geht die Steuerung an den EXLST-Ausgang NOFIND. Auch wenn die Datei bereits auf den Satz positioniert ist, der den LIMIT-Schlüssel enthält, geht die Steuerung an den EXLST-Ausgang NOFIND; die Dateiposition wird nicht verändert. Ist der über KEYARG adressierte Schlüssel kleiner als die aktuelle Zeigerposition (bzw. größer bei REVERSE=YES), geht die Steuerung an den EXLST-Ausgang USERERR.

LOGTEST
gibt an, ob bei der Suche über die logische Markierung nur Sätze bereitgestellt werden sollen, die alle oder mindestens eine der in der Bit-Maske gesetzten Bedingungen erfüllen. Die Bit-Maske muss in dem Feld enthalten sein, auf das der FCB-Operand KEYARG verweist. Es muss mindestens ein Bit der Maske gesetzt sein sonst geht die Steuerung an den EXLST-Ausgang USERERR.

= ANY
liest den nächsten Satz, in dessen logischer Markierung mindestens ein Bit gesetzt ist, das auch in der Maske gesetzt ist.

= ALL
liest den nächsten Satz, in dessen logischer Markierung alle Bits gesetzt sind, die den in der Maske gesetzten Bits entsprechen.

MF = (E,...)
erzeugt den Systemaufruf. Die Operandenliste, die mit MF=L generiert wurde, wird für die Ausführung des Makroaufrufs ausgewertet.

= E,addr
Adresse der mit MF=L erzeugten Operandenliste. Sollen die Adressen von FCB und „area“ in Registern übergeben werden, müssen diese vor Ausführung des Makroaufrufs mit der gültigen Adressen geladen werden.

= E,r
Die Adresse der mit MF=L erzeugten Operandenliste steht im Register „r“.

MF = L
nur bei PARMOD=24
Es wird eine 8 Byte lange Operandenliste generiert, der Makroaufruf wird nicht ausgeführt. Die Operandenliste ist auf Wortgrenze ausgerichtet und enthält:
– das Operandenbyte 1 (siehe unten, Tabelle GETFL-1)

– die FCB-Adresse oder die Nummer des Registers, das die FCB-Adresse enthält (=1)
– das Operandenbyte 2 (siehe unten, Tabelle GETFL-2)
– die Adresse des Bereichs, in den der Satz übertragen wird, oder die Nummer des Registers, das beim Makroaufruf mit MF=E diese Adresse enthält.

Die Operandenliste muss symbolisch adressierbar sein (= symbolische Adresse des Makroaufrufs).

PARMOD
Gibt den Generierungsmodus an, entsprechend dem im FCB-Makroaufruf für die Datei gültigen Wert.

Abhängig von MF=L und PARMOD-Angabe werden unterschiedliche „Operandenlisten“ generiert.

Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der generierte Objektcode ist nur im 16-MB-Adressraum ablauffähig. Bei MF=L wird eine 8-Byte-Operandenliste generiert, ohne MF=L werden die Parameter in die Register 0 und 1 übernommen.

= 31
Der generierte Objektcode ist im 2-GB-Adressraum ablauffähig. Die Informationen zur Makroausführung sind im FCB enthalten.

REVERSE = YES
Die Verarbeitung verläuft „rückwärts“ in Richtung Dateianfang.

Voreinstellung: die Dateiverarbeitung verläuft „vorwärts“ in Richtung Dateiende

VALTEST
gibt an, in welcher Relation die Wertmarkierung des zu lesenden Satzes zu der Vorgabe stehen soll, auf die der FCB-Operand verweist.

= GT
(= greater than) der Wert im Datensatz muss größer sein als die Wertvorgabe im KEYARG-Feld

= GE
(= greater or equal) der Wert im Datensatz ist größer oder gleich der Wertvorgabe im KEYARG-Feld

= EQ
(= equal) die beiden Werte müssen übereinstimmen

= NE
(= not equal) die beiden Werte dürfen nicht übereinstimmen

= LE
(= less or equal) der Wert im Datensatz ist kleiner oder gleich der Wertvorgabe im KEYARG-Feld

= LT
(= less than) der Wert im Datensatz ist kleiner als die Wertvorgabe im KEYARG-Feld.

Operandenliste zu MF=L

Wort 1		**Wort 2**	
Op-Byte 1	FCB-Adresse oder Register-nummer	Op-Byte 2	Area-Adresse oder Register-nummer

Op-Byte 1 = Operanden-Byte 1
verschlüsselte GETFL-Operanden und LOGTEST (siehe Tabelle „Operanden-Byte 1“ auf Seite 688.

FCB-Adresse/Register
hier ist entweder die Adresse des FCB oder ein Register anzugeben (rechtsbündig ausgerichtet), das die Adresse des FCB enthält.

Op-Byte 2 = Operanden-Byte 2
verschlüsselte Informationen zu den GETFL-Operanden LOCK/NOLOCK, fcbadr, area (siehe Tabelle „Operanden-Byte 2“ auf Seite 688.

Area-Adresse/Register
hier ist entweder die Adresse des Bereichs oder ein Register anzugeben (rechtsbündig ausgerichtet), das die Adresse des Bereichs enthält, in den der Satz übertragen werden soll.

Operanden-Byte 1

Bitposition/Bitmuster								Bedeutung
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
0	0	1	0					VALTEST=GT
0	1	0	0					VALTEST=LT
1	0	0	0					VALTEST=EQ
0	1	1	1					VALTEST=NE
1	0	1	0					VALTEST=GE
1	1	0	0					VALTEST=LE
0	0	0	0					VALTEST=0 oder ungültig
				1 0				LOGTEST-Operand angegeben LOGTEST-Operand null oder ungültig
					1 0			LOGTEST=ALL LOGTEST=ANY
						1 0		LIMIT=KEY LIMIT=END
							1 0	REVERSE=YES REVERSE=null

Operanden-Byte 2

Bitmuster/Bitposition (3-0 nicht verwendet)				Bedeutung	
7	**6**	**5**	**4**	**PARMOD=24**	**PARMOD=31**
1 0				LOCK angegeben oder Standardeinstellung NOLOCK angegeben	LOCK angegeben oder Standardeinstellung NOLOCK angegeben
	1 0			FCB-Adresse in Register enthalten FCB-Adresse angegeben	 - nicht verwendet -
		1 0		'area' -Adresse nicht angegeben 'area' -Adresse angegeben	 - nicht verwendet -
			1 0	'area' -Adresse in Reg. 0 enthalten 'area' -Adresse angegeben	 - nicht verwendet -

Hinweis zur Programmierung

Der GETFL-Makroaufruf zerstört die Register 0, 1, 14 und 15.

Übersicht über die EXLST-Ausgänge

EXLST-Ausgang	betroffener GEFTL-Operand	Bedeutung
EOFADDR	LIMIT = END	Kein passender Satz vorhanden
NOFIND	LIMIT = KEY	– kein passender Satz im definierten Bereich vorhanden – Grenze ist identisch mit der Zeigerposition
USERERR	---- LIMIT = KEY LOGTEST	Datei wurde nicht mit Markierungen erstellt oder anderer Anwenderfehler (wie z.B. falscher OPEN-Modus) Grenze ist bereits überschritten Die Logikmaske enthält nur Nullen: kein Bit gesetzt

GETKY – Satz mit angegebenem Schlüssel lesen

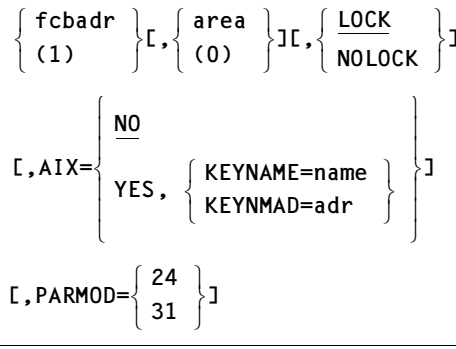
Makrotyp: R bei PARMOD=24
0 bei PARMOD=31

Der GETKY-Makroaufruf liest einen Satz, bei dem der Primär- bzw. ein Sekundärschlüssel mit dem im KEYARG-Feld bereitgestellten Wert übereinstimmt. Das KEYARG-Feld wird über den KEYARG- Operanden des FCB-Makroaufrufs adressiert.

Wird kein Satz mit dem angegebenen Schlüsselwert gefunden, wird dem Anwenderprogramm über die NOFIND-Adresse (siehe Makro EXLST, Operand NOFIND, Seite 401) die Steuerung übergeben. Der Zeiger des Primär- bzw. des Sekundärschlüssels wird auf den gesuchten Wert gesetzt.

Enthält die Datei mehrere Sätze mit gleichen Werten für den im GETKY-Makroaufruf angesprochenen Primär- bzw. Sekundärschlüssel, wird der Erste dieser Sätze bereitgestellt bzw. der Satz, auf dessen Primärschlüssel im zugehörigen Sekundärindexblock als Erstes verwiesen wird.

Format

Operation	Operanden
GETKY	

Operandenbeschreibung

fcbadr
Adresse des FCB, der zu verarbeitenden Datei.

Wenn die Datei über einen Sekundärschlüssel gelesen werden soll, muss dieser FCB in seiner 31-Bit-Schnittstelle vorliegen.

(1)
Die FCB-Adresse steht im Register 1.

area
Adresse des Feldes, in das der Satz gebracht werden soll; im Locate-Mode wird „area“ ignoriert.

(0)
Die Adresse des Feldes, in das der Satz gebracht werden soll, steht im Register 0.

LOCK
Die Satz- oder Datenblocksperre bleibt nach Ausführung des Makroaufrufs erhalten (explizite Sperre).

NOLOCK
Es wird keine explizite Sperre gesetzt.

AIX
Gibt an, ob der Satz über seinen Primär- oder einen Sekundärschlüssel bereitgestellt werden soll.

= NO
Der Satz wird über seinen Primärschlüssel bereitgestellt (Voreinstellung).

= YES
Kann nur angegeben werden, wenn
- *die 31-Bit-Schnittstelle des Makros generiert wird (über den Operanden PARMOD=31 oder den Makroaufruf GPARMOD 31) und*
- *der Makro sich auf einen 31-Bit-FCB bezieht.*

Der Satz wird über den im Operanden KEYNAME oder KEYNMAD vereinbarten Sekundärschlüssel bereitgestellt.

KEYNAME = name
Gibt den Namen des Sekundärschlüssels an, über den der Satz gelesen werden soll. name muss der Name eines für die aktuelle Datei vereinbarten Sekundärschlüssels sein. Die Namen aller für eine Datei definierten Sekundärschlüssel kann mit dem Makro SHOWAIX oder dem Kommando SHOW-INDEX-ATTRIBUTES ermittelt werden.

KEYNMAD = adr
Gibt die symbolische Adresse (den Namen) eines Feldes an, in dem der Anwender den Namen des Sekundärschlüssels hinterlegt hat, über den der Satz gelesen werden soll. Das Feld mit der symbolischen Adresse adr muss zum Zeitpunkt der Makroausführung den Namen eines für die aktuelle Datei vereinbarten Sekundärschlüssels enthalten.

PARMOD
Gibt den Generierungsmodus an.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24
Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (nur 24-Bit-Adressierung).

= 31
Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist (24-Bit- oder 31-Bit-Adressierung).

Hinweis zur Programmierung

Der GETKY-Makroaufruf zerstört die Register 0, 1, 14 und 15.

GETR – Sequenziell „rückwärts“ lesen

Makrotyp: R bei PARMOD=24
0 bei PARMOD=31

Der GETR-Makroaufruf liest den nächsten Satz der Datei in Richtung Dateianfang; die Datei wird rückwärts gelesen.

Wird ein Satz angefordert, der außerhalb der Datei liegt, wird dem Anwender über EOFADDR (siehe Makro EXLST, Operand EOFADDR, Seite 397) die Steuerung übertragen.

Das Programm kann jederzeit von GET auf GETR umschalten und umgekehrt. Vor dem Umschalten muss die Datei nicht auf Dateianfang oder -ende positioniert werden.

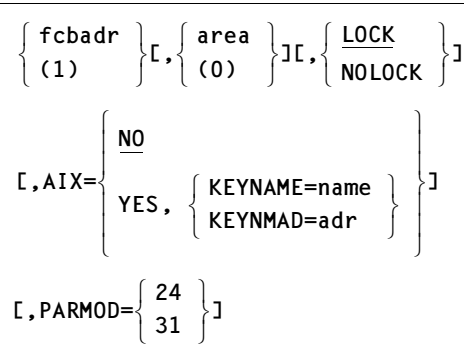
Folgt auf einen GET-Makroaufruf, der einen Satz mit dem Primär- bzw. Sekundärschlüsselwert K_n bereitgestellt hat, ein GETR-Aufruf, der sich ebenfalls auf den Primär- bzw. den gleichen Sekundärschlüssel bezieht, so liefert dieser GETR-Aufruf den Satz mit dem nächstniedrigeren Primär- bzw. Sekundärschlüsselwert K_{n-1} ($K_{n-1} < K_n$).

Enthält eine Datei Sätze mit gleichen Primärschlüsselwerten, so werden bei GETR die Sätze in der umgekehrten Reihenfolge geliefert, in der sie in die Datei eingefügt wurden. D.h. als ersten Satz einer Gruppe von Sätzen mit gleichem Schlüssel erhält man den Satz, der als Letzter in die Datei eingefügt wurde.

Wird die Datei über einen Sekundärschlüssel gelesen und enthält sie Datensätze mit gleichen Werten für diesen Sekundärschlüssel, so stellt GETR diese Sätze in der umgekehrten Reihenfolge bereit, in der die Sekundärschlüsselwerte entstanden sind.

Folgt ein GETR-Makroaufruf auf einen SETL KEY, so wird der Satz zur Verfügung gestellt, auf den mit SETL KEY positioniert wurde.

Format

Operation	Operanden
GETR	

Operandenbeschreibung

fcbadr
Adresse des FCB für die zu verarbeitende Datei.
Wenn die Datei über einen Sekundärschlüssel gelesen werden soll, muss dieser FCB in seiner 31-Bit-Schnittstelle vorliegen.

(1)
Die FCB-Adresse steht im Register 1.

area
Adresse des Feldes, in das der Satz gebracht werden soll; im Ortungsbetrieb wird „area“ ignoriert.

(0)
Die Adresse des Feldes, in das der Satz gebracht werden soll, steht im Register 0.

LOCK
Die Satz- oder Datenblocksperre soll nach Ausführung des Makroaufrufs erhalten bleiben (explizite Sperre).

NOLOCK
Es wird keine explizite Sperre gesetzt.

AIX
Gibt an, ob der Satz über seinen Primär- oder einen Sekundärschlüssel bereitgestellt werden soll.

= NO
Der Satz wird über seinen Primärschlüssel bereitgestellt (Voreinstellung).

= YES
Kann nur angegeben werden, wenn
– die 31-Bit-Schnittstelle des Makros generiert wird (über den Operanden PARMOD=31 oder den Makroaufruf GPARMOD 31) und
– der Makro sich auf einen 31-Bit-FCB bezieht.

Der Satz wird über den im Operanden KEYNAME oder KEYNMAD vereinbarten Sekundärschlüssel bereitgestellt.

KEYNAME = name
Gibt den Namen des Sekundärschlüssels an, über den der Satz gelesen werden soll. name muss der Name eines für die aktuelle Datei vereinbarten Sekundärschlüssels sein. Die Namen aller für eine Datei definierten Sekundärschlüssel kann mit dem Makro SHOWAIX oder dem Kommando SHOW-INDEX-ATTRIBUTES ermittelt werden.

KEYNMAD = adr
Gibt die symbolische Adresse (den Namen) eines Feldes an, in dem der Anwender den Namen des Sekundärschlüssels hinterlegt hat, über den der Satz gelesen werden soll. Das Feld mit der symbolischen Adresse adr muss zum Zeitpunkt der Makroausführung den Namen eines für die aktuelle Datei vereinbarten Sekundärschlüssels enthalten.

PARMOD
Gibt den Generierungsmodus an.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24
Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (nur 24-Bit-Adressierung).

= 31
Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist (24-Bit- oder 31-Bit-Adressierung).

Hinweis zur Programmierung

Der GETR-Makroaufruf zerstört die Register 0, 1, 14 und 15.

IDBPL – BTAM-Operandenliste mit symbolischen Namen versorgen

Makrotyp: O-Typ

Der Makroaufruf IDBPL generiert eine DSECT, die die Felder des BTAM-Aktionsmakroaufrufs mit symbolischen Namen versieht.

Format

Operation	Operanden
IDBPL	[D][,{prefix / *}][,PARMOD={24 / 31}]

Operandenbeschreibung

D
Gibt an, dass eine DSECT-Anweisung generiert werden soll. Wird „D“ nicht angegeben, wird keine DSECT-Anweisung generiert.

prefix
Präfix (1 Zeichen), mit dem jeder symbolische Name beginnt.

Voreinstellung: jedem Namen wird der Buchstabe „I“ vorangestellt

*****
den Namen wird kein Präfix vorangestellt

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.

Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst.
Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

IDFCB – FCB mit symbolischen Namen versorgen

Makrotyp: O-Typ

Der Makroaufruf IDFCB generiert eine DSECT für den Dateisteuerblock (FCB), sodass der Benutzer, bei entsprechender Initialisierung eines Basisregisters, alle Felder des FCB symbolisch adressieren kann.

Format

Operation	Operanden
IDFCB	[D][,{prefix / *}][,PARMOD={24 / 31}]

Operandenbeschreibung

D
gibt an, dass eine DSECT-Anweisung generiert werden soll; standardmäßig generiert das System den Makroaufruf nicht als DSECT.

prefix
Präfix (1 Zeichen), das allen Namen der DSECT vorangestellt werden soll; standardmäßig werden DSECTs mit dem Präfix „I“ generiert.

*****
Es wird kein Präfix verwendet.

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.
Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst. Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig. Es wird eine DSECT für den „alten“ FCB (BS2000 ≤ V8.5) generiert, d.h. für die FCB-Erweiterung muss wie bisher mit dem Makroaufruf IDFCBE eine DSECT generiert werden (siehe FCB-Makro, Seite 410 und IDFCBE-Makro, Seite 698).

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert. Für den neuen FCB (BS2000 ≥ V9.0) gibt es keine FCB-Erweiterung mehr, es wird daher auch kein IDFCBE-Makroaufruf benötigt.

IDFCBE – FCBE mit symbolischen Namen versorgen

Makrotyp: O-Typ

Der Makroaufruf IDFCBE generiert eine DSECT für die FCB-Erweiterung des 24-Bit-TU-FCBs. Bei entsprechender Initialisierung eines Basisregisters kann der Benutzer die Felder der FCB-Erweiterung symbolisch adressieren.

Diese FCB-Erweiterung wird nur dann angelegt, wenn der 24-Bit-FCB generiert wurde und bei Bandverarbeitung die Operanden BUFOFF, FSEQ oder LABEL in FILE- oder FCB-Makroaufruf angegeben wurden oder wenn bei Plattenverarbeitung der Operand OPTION= GLODEF angegeben wurde. Ist dies nicht der Fall, wird auch kein IDFCBE-Makroaufruf benötigt.

Um die Anfangsadresse der FCBE zu ermitteln, wird außer dem IDFCBE-Makroaufruf auch der IDFCB-Makroaufruf (mit PARMOD=24!) benötigt.

Da der 31-Bit-FCB keine FCB-Erweiterung (FCBE) mehr hat, wird in XS-Umgebung der IDFCBE-Makroaufruf nicht unterstützt.

Format

Operation	Operanden
IDFCBE	[D][,{ prefix / * }]

Operandenbeschreibung

D
gibt an, dass eine DSECT generiert werden soll; standardmäßig wird keine DSECT generiert.

prefix
Präfix (1 Zeichen), das allen Namen der DSECT vorangestellt wird, standardmäßig der Buchstabe „I“.

*****
Es wird kein Präfix verwendet.

IDPPL – PAM-Operandenliste mit symbolischen Namen versorgen

Makrotyp: O-Typ

Der Makroaufruf IDPPL wird dazu benutzt, einen Pseudoabschnitt (DSECT) zu generieren, der die einzelnen Felder des PAM-Makroaufrufs mit symbolischen Namen versieht.

Format

Operation	Operanden
IDPPL	`[D][,{prefix / *}][,PARMOD={24 / 31}]`

Operandenbeschreibung

D
Gibt an, dass eine DSECT-Anweisung generiert werden soll.

Standardmäßig wird nicht eine DSECT-, sondern eine CSECT-Anweisung generiert.

prefix
Gibt das Präfix an, d.h., das Zeichen, mit dem jeder symbolische Name beginnen soll.

*****
Gibt an, dass kein Präfix verwendet werden soll.

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.

Standardwert: der durch einen GPARMOD-Aufruf oder durch den Assembler voreingestellte Wert

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst.
Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

IMPNFIL – Katalogeintrag für Node-Files erstellen (importieren)

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Der IMPNFIL-Makroaufruf katalogisiert auf Net-Storage-Volumes gespeicherte Node-Files, für die der aufrufende Auftrag das Eigentumsrecht hat. Das DVS legt den Katalogeintrag für ein Node-File im TSOSCAT und im Dateikatalog des Net-Storage-Volumes anhand der Inode-Attribute auf dem NFS-Server an.

Mit Angabe eines teilqualifizierten Dateinamens oder mit Wildcards kann der Benutzer mit einem Aufruf auch mehrere Dateien importieren.

Hinweise

– Mit-Eigentümer einer Benutzerkennung dürfen unter dieser Kennung Node-Files importieren.

– Wenn Einträge im Benutzerkatalog ersetzt werden müssen (REPLACE= *YES/*NFU), dürfen diese nicht gesperrt sein und Schreibzugriff muss zulässig sein.

Format

Operation	Operanden
IMPNFIL	,VOLUME=<c-string: 1..6> / <var: char:6> ,FILENAM=<c-string 1..80>:<filename 1..54 with-wild-without-cat(80)> / <var: char:80> ,PUBSET=***STD** / <c-string: 1..4> / <var: char:4> ,FILESTR=***STD** / *PAM / <var: enum-of_filestr_s: 1> ,REPLACE=***NO** / *YES / *NFU / <var: enum-of_replace_s: 1> ,IGNPROT=***NO** / *YES / <var: enum-of_ignprot_s: 1> ,LIST=***NO** / *SYSOUT / *SYSLST / *BOTH / <var: enum-of_list_s: 1> ,REPORT=***ERROR** / *FULL / <var: enum-of_report_s: 1> ,EQUATES=***YES** / *NO MF=L
	MF=D,PREFIX=**D** / <pre>
	MF=E,PARAM=<name 1..27>
	MF=C / M **,PREFIX=D / <pre>** **,MACID=MAN / <macid>**

Operandenbeschreibung

VOLUME
Archivnummer (VSN) des Net-Storage-Volumes, auf dem die zu importierenden Node-Files gespeichert sind.

=<c-string: 1..6>
VSN des Net-Storage-Volumes.

=<var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 6 Byte, in dem die VSN des Net-Storage-Volumes abgelegt ist.

FILENAM
Auswahl der Node-Files, die importiert werden sollen.

=<c-string 1..80: filename 1..54 with-wild-without-cat(80)>
Pfadname des Node-Files auf dem Net-Storage-Volume. Eine Katalogkennung darf nicht angegeben werden. Die Wildcard-Angabe (Musterzeichen) ermöglicht die Auswahl einer Dateimenge.

Der nicht privilegierte Benutzer kann nur Dateien seiner Benutzerkennung importieren. Der privilegierte Benutzer (Privileg TSOS) kann auch Dateien anderer Benutzer importieren. Die Angabe von Wildcards in der Benutzerkennung ist erlaubt.

=<var: char:80>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 80 Byte, in dem der Pfadname bzw. die Musterzeichenfolge für die gewünschte(n) Datei(en) abgelegt ist.

PUBSET
Bestimmt den Pubset, in dem die Dateien katalogisiert werden sollen. Das im Operanden VOLUME angegebene Net-Storage-Volume muss dem hier angegebenen Pubset zugeordnet sein.

=*STD
Die Katalogeinträge werden im Dateikatalog des Default-Pubsets der Benutzerkennung eingerichtet.

=<c-string: 1..4>
Pubset-Id des Pubsets. Die Katalogeinträge werden im Dateikatalog des angegebenen Pubsets eingerichtet.

=<var: char:4>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 4 Byte, in dem die Pubset-Id abgelegt ist.

FILESTR
Bestimmt das Attribut FILE-STRUCTURE des Node-Files, das in Abhängigkeit vom Operanden REPLACE in den Dateikatalog eingetragen wird.

=*STD
Bei REPLACE=*NO/*YES gilt: Ein Node-File wird als PAM-Datei ins BS2000 importiert, wenn die Dateigröße auf dem NFS-Dateisystem ungleich Null ist. Wenn die Dateigröße auf dem NFS-Dateisystem gleich Null ist, so erhält die importierte Datei die Standard-Attribute einer mit CREATE-FILE erzeugten Datei.
Bei REPLACE=*NFU gilt: Die Katalogeinträge der Node-Files werden im BS2000 unabhängig von dem Attribut FILE-STRUCTURE aktualisiert.

=*PAM
Bei REPLACE=*NO/*YES gilt: Ein Node-File wird unabhängig von der Dateigröße auf dem NFS-Dateisystem als PAM-Datei ins BS2000 importiert.
Bei REPLACE=*NFU werden die Katalogeinträge von PAM-Node-Files im BS2000 aktualisiert.

=<var: enum-of_filestr_s: 1>
Name des Feldes mit dem Wert für FILESTR.

REPLACE
Gibt an, ob im BS2000 bereits existierende Dateien ersetzt werden oder ob nur der Katalogeintrag anhand der Inode-Attribute auf dem NFS-Server aktualisiert wird.

=*NO
Bereits existierende Dateien werden weder ersetzt noch ihre Katologeinträge aktualisiert.

=*YES
Bereits existierende Dateien auf dem Pubset werden durch die angegebenen Node-Files ersetzt. Dabei werden Dateien auf Public-Space und auf Net-Storage gelöscht, und Dateien auf Privatplatte werden exportiert. Beim Importieren der Node-Files werden werden die Einträge im TSOSCAT und im Dateikatalog des Net-Storage-Volumes neu erstellt.

=*NFU
Bei bereits existierenden Dateien werden die Einträge im TSOSCAT und im Dateikatalog des Net-Storage-Volumes anhand der Inode-Attribute auf dem NFS-Server aktualisiert. Dabei bestimmt der Operand FILESTR, dass die Aktualisierung der Katalogeinträge nur für Dateien mit der angegebenen Dateistruktur erfolgt. Mit FILESTR=*STD werden die Dateien unabhängig von der Dateistruktur aktualisiert.

=<var: enum-of_replace_s: 1>
Name des Feldes mit dem Wert für REPLACE.

IGNPROT
Der Operand steht nur dem privilegierten Benutzer (Privileg TSOS) zur Verfügung.
Gibt an, ob bereits katalogisierte Dateien ohne Beachtung eines bestehenden Schreibschutzes überschrieben werden sollen.

=*NO
Der Schreibschutz wird beachtet.

=*YES
Der Schreibschutz wird ignoriert.

=<var: enum-of_ignprot_s: 1>
Name des Feldes mit dem Wert für IGNPROT.

LIST
Gibt an, ob ein Verarbeitungsprotokoll nach SYSOUT und/oder SYSLST ausgegeben werden soll. Voreingestellt ist *NONE, d.h. es wird kein Protokoll erstellt.

=*NO
Es erfolgt keine Ausgabe.

=*SYSOUT
Das Verarbeitungsprotokoll wird nach SYSOUT ausgegeben.

=*SYSLST
Das Verarbeitungsprotokoll wird nach SYSLST ausgegeben.

=*BOTH
Das Verarbeitungsprotokoll wird nach SYSOUT und SYSLST ausgegeben.

=<var: enum-of_list_s: 1>
Name des Feldes mit dem Wert für LIST.

REPORT
Bestimmt den Umfang des Protokolls, wenn im Operanden LIST ein Verarbeitungsprotokoll angefordert wurde.

=ERROR
Es werden nur Dateien aufgelistet, die nicht importiert werden konnten. Die Ursache wird jeweils mit einem Meldungsschlüssel angezeigt.

=*FULL
Es werden alle Dateien aufgelistet. Für die nicht importierbaren Dateien wird die Ursache jeweils mit einem Meldungsschlüssel angezeigt.

=<var: enum-of_report_s: 1>
Name des Feldes mit dem Wert für REPORT.

EQUATES
Steuerungs-Operand nur für MF=C und MF=D:
Gibt an, ob bei der Expansion des Parameterbereichs auch Equates für die Werte der Felder des Parameterbereichs generiert werden sollen.

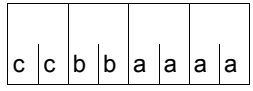
= *YES
Bei der Expansion des Parameterbereichs werden auch Equates für die Werte der Felder des Parameterbereichs generiert.

= *NO
Bei der Expansion des Parameterbereichs werden keine Equates für die Werte der Felder des Parameterbereichs generiert.

Returncodes

Der Returncode wird im Standardheader des Parameterbereichs abgelegt. Der Parameterbereich darf dann nicht im Read-only-Bereich liegen, sonst erfolgt Programmterminierung.

Standard-header:



Über die Ausführung des Makros IMPNFIL wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2,bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'00'	X'0000'	kein Fehler
X'00'	X'40'	X'0501'	CMS oder FILE: angeforderter Katalog nicht verfügbar
X'00'	X'40'	X'0512'	Pubsetkennung ist nicht im MRSCAT eingetragen
X'00'	X'40'	X'051B'	Benutzerkennung im angegebenen Pubset unbekannt
X'00'	X'40'	X'051C'	Benutzer hat kein Zugriffsrecht auf angegebenen Pubset
X'00'	X'40'	X'0535'	Es besteht keine Zugriffsberechtigung auf den Katalogeintrag der Datei
X'00'	X'20'	X'0578'	interner Fehler bei Dateischutzprüfung
X'00'	X'82'	X'0594'	nicht genügend virtueller Speicher verfügbar
X'00'	X'20'	X'05C7'	interner Fehler im DMS
X'00'	X'01'	X'05EE'	Pfadname nach Komplettierung zu lang
X'00'	X'40'	X'05FC'	angegebene Benutzerkennung nicht im Home-Pubset
X'00'	X'40'	X'0610'	Mindestens für einen der ausgewählten Dateinamen lieferte die Funktionsausführung einen Returncode
X'00'	X'01'	X'0624'	Dateiname ungültig
X'00'	X'40'	X'0640'	Zugriff auf Net-Storage wird vom Subsystem ONETSTOR wegen Kommunikationsproblemen mit dem Net-Client abgewiesen
X'00'	X'04'	X'0642'	große Dateien auf Pubset nicht erlaubt
X'00'	X'40'	X'0643'	Net-Client meldet Zugriffsfehler
X'00'	X'40'	X'0644'	Net-Client meldet internen Fehler
X'00'	X'40'	X'0645'	Datei auf Net-Storage nicht vorhanden
X'00'	X'40'	X'0649'	Net-Server meldet ACL-Fehler
X'00'	X'40'	X'064A'	Net-Client meldet, Zugriff auf Dateien auf dem Net-Storage-Volume verboten ist
X'00'	X'40'	X'064B'	Zugriff auf Node-Files vom Net-Client nicht unterstützt

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'40'	X'064C'	Verzeichnis der angegebenen Benutzerkennung existiert nicht auf Net-Server
X'00'	X'40'	X'064D'	Datei ist kein Node-File
X'00'	X'40'	X'064E'	Node-File liegt nicht auf dem angegebenen Net-Storage-Volume
X'00'	X'40'	X'064F'	FCB-TYPE der Datei und angegebene Dateistruktur stimmen nicht überein
X'00'	X'40'	X'0650'	kein Node-File gefunden, das importiert oder aktualisiert werden kann
X'00'	X'40'	X'0651'	Datei existiert, Import nicht möglich
X'00'	X'40'	X'06CC'	nur bei Auswahlangabe (Wildcard): Keine Datei entspricht der Auswahlangabe
X'00'	X'01'	X'FFFF'	falsche Funktionsnummer im Standardheader
X'00'	X'03'	X'FFFF'	falsche Versionsnummer im Standardheader

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Layout der Operandenliste

Makroauflösung mit MF=D, sowie Standardwerten für EQUATES, PREFIX und MACID:

```
IMPNFIL MF=D
        MFTST MF=D,PREFIX=D,MACID=MAN,ALIGN=F,
              DMACID=MAN,SUPPORT=(E,D,C,M,L),DNAME=MANGLPL
DMANGLPL DSECT ,
              *,##### PREFIX=D, MACID=MAN #####
*   PARAMETER AREA
DMANHDR  FHDR  MF=(C,DMAN),EQUATES=NO
DMANHDR  DS    0A
DMANFHE  DS    0XL8           0   GENERAL PARAMETER AREA HEADER
*
DMANIFID DS    0A             0   INTERFACE IDENTIFIER
DMANFCTU DS    AL2            0   FUNCTION UNIT NUMBER
*                                BIT 15    HEADER FLAG BIT,
*                               MUST BE RESET UNTIL FURTHER NOTICE
*                               BIT 14-12 UNUSED, MUST BE RESET
*                               BIT 11-0  REAL FUNCTION UNIT NUMBER
DMANFCT  DS    AL1            2   FUNCTION NUMBER
DMANFCTV DS    AL1           3   FUNCTION INTERFACE VERSION NUMBER
*
DMANRET  DS    0A             4   GENERAL RETURN CODE
DMANSRET DS    0AL2           4   SUB RETURN CODE
DMANSR2  DS    AL1            4   SUB RETURN CODE 2
DMANSR1  DS    AL1            5   SUB RETURN CODE 1
DMANMRET DS    0AL2           6   MAIN RETURN CODE
DMANMR2  DS    AL1            6   MAIN RETURN CODE 2
DMANMR1  DS    AL1            7   MAIN RETURN CODE 1
DMANFHL  EQU   8             8   GENERAL OPERAND LIST HEADER LENGTH
*
DMANVOLUM DS    CL6                    VOLUME
DMANFNAME DS    CL80                   FILENAME
DMANPUBID DS    CL4                    PUBSET
DMANFILS DS    FL1                   FILESTRUC
*   FILESTRUC VALUES
DMANSTDF EQU   0                    FILESTRUC = STD
DMANPAMF EQU   1                    FILESTRUC = PAM
DMANSAMF EQU   2                    FILESTRUC = SAM
*
DMANREPL DS    FL1                   REPLACE
*   REPLACE VALUES
DMANREPN EQU   0                    REPLACE=NO
DMANREPY EQU   1                    REPLACE=YES
DMANREPU EQU   2                   REPLACE=NODE FILE
*
DMANIGNP DS    FL1                   IGNPROT
*   IGNPROT VALUES
```

```
DMANIGNO EQU   0                       IGNPROT = NO
DMANIGYE EQU   1                       IGNPROT = YES
*
DMANLIST DS    FL1                     LIST
*   LIST VALUES
DMANLISN EQU   0                       LIST = NO
DMANOUTO EQU   1                       LIST = SYSOUT
DMANOUTL EQU   2                       LIST = SYSLST
DMANOUTB EQU   3                       LIST = BOTH
*
DMANREPO DS    FL1                     REPORT
*   REPORT VALUES
DMANREPE EQU   0                       REPORT = NO
DMANREPF EQU   1                       REPORT = FULL
*
DMANRES1 DS    XL5                     ALIGNMENT
DMAN#    EQU   *-DMANHDR
```

Beispiel für eine Aufruffolge

```
MVC   IMPNMFC(DMAN#),IMPNMFL
         IMPNFIL MF=M,                                                  -
               PARAM=IMPNMFC,                                           -
               VOLUME='P@BXOO',FILENAM='*',                             -
               PUBSET='X'
         IMPNFIL MF=E,PARAM=IMPNMFC
             .
             .
IMPNMFC  IMPNFIL MF=C
IMPNMFL  IMPNFIL MF=L,VOLUME='*DUMMY',FILENAM='AAA'
```

IMPORT – Katalogeintrag für Dateien erstellen (importieren)

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form) (siehe Seite 870)

Der IMPORT-Makroaufruf katalogisiert auf Privatplatten oder Net-Storage-Volumes gespeicherte Dateien, für die der aufrufende Auftrag das Eigentumsrecht hat. Das DVS übernimmt die Dateimerkmale aus dem F1-Kennsatz der Privatplatte bzw. aus dem Katalog des Net-Storage-Volumes in den Katalogeintrag. Es kann teilqualifizierte Dateinamen verarbeiten, sodass der Benutzer mit einem Aufruf mehrere Dateien importieren kann.

Beim Einbringen von Dateigenerationsgruppen mit Generationen auf verschiedenen Platten ist zu beachten, dass Generationen nur dann katalogisiert werden, wenn der Gruppeneintrag im System-Katalog vorhanden ist oder auf der ersten zu importierenden Platte steht. Andernfalls fehlen später die Katalogeinträge der Generationen, die vor dem Gruppeneintrag importiert werden. Sie müssen nachträglich durch einen IMPORT- oder FILE-Makroaufruf (Operand STATE=FOREIGN) katalogisiert werden.

Die Funktionen IMPORT und ERASE (Operanden CATALOG bzw. DELETE-OR-EXPORT und VOLUME) sind nicht exakt gegensätzlich: Beim Exportieren eines Datenträgers löscht das DVS die Katalogeinträge aller Dateien, die auf diesem Datenträger Speicherplatz belegen. Wird derselbe Datenträger wieder importiert, so erstellt das DVS nur Katalogeinträge für die Dateien, die auf diesem Datenträger beginnen (d.h. Dateien, die bei der Primärzuweisung Speicherplatz auf dieser Platte erhielten).

Hinweise

– Mit-Eigentümer einer Benutzerkennung dürfen unter dieser Kennung permanente Dateien anlegen.

– Aus dem F1-Kennsatz bzw. aus dem Katalog des Net-Storage-Volumes können gesperrte Einträge importiert werden. Müssen jedoch Einträge im Benutzerkatalog ersetzt werden (REPLACE= YES/ABS), dürfen diese nicht gesperrt sein und Schreibzugriff muss zulässig sein.

Format

Operation	Operanden
IMPORT	[pfadname],VOLUME=vsn,DEVICE=gerät [,AREA=(adr,länge)][,REPLACE={NO / YES / ABS}] [,GEN={YES / NO}][,LIST={YES / NO / ONLY}] [,PVSID=catid] [,NUSERID=userid] [,MF=L][,VERSION=1]
	MF=(E,{adr / (r)})[,VERSION=1]
	MF={D / C}[,PREFIX={pre / *}][,VERSION=1]

Operandenbeschreibung

AREA
legt den Ausgabebereich für den IMPORT-Makro fest. Der Operand AREA kann weggelassen werden, wenn LIST=NO angegeben wird.

= (adr,länge)
adr symbolische Adresse des Ausgabebereichs
länge Länge des Ausgabebereichs

DEVICE = gerät
Bezeichnet den Gerätetyp, auf dem der Datenträger bereitzustellen ist; mögliche Angaben für „gerät“ siehe die Gerätetabelle im Handbuch „Systeminstallation [16].
Die ab BS2000-Version 9.5 neu eingeführten Gerätetypen werden nur im Zusammenhang mit VERSION=1 unterstützt. Für Net-Storage-Volumes ist anstelle des Gerätetyps der Volumetyp NETSTOR anzugeben.

Jede Angabe eines dem System bekannten Plattengerätetyps wird wie die Angabe STDDISK behandelt.

GEN
Für Dateigenerationsgruppen:
GEN legt fest, ob nur der Gruppeneintrag oder auch die auf derselben Privatplatte gespeicherten Dateigenerationen katalogisiert werden.

= YES
Steht der Gruppeneintrag auf der Privatplatte, katalogisiert das DVS die FGG und alle zu ihr gehörenden Generationen, die auf dieser Platte beginnen. Gibt es weder auf der Platte noch im Benutzerkatalog einen Gruppeneintrag, werden keine Dateigenerationen katalogisiert.

= NO
Das DVS übernimmt nur den Gruppeneintrag der FGG.

LIST
Legt fest, wie die Makroverarbeitung protokolliert wird (siehe „Hinweis zur Programmierung“ auf Seite 714).

= YES
Die Makroverarbeitung wird protokolliert.

= NO
Es werden keine Rückinformationen über die Makroverarbeitung ausgegeben.

= ONLY
Bewirkt, dass der IMPORT-Makroaufruf nicht ausgeführt, sondern nur „simuliert“ wird, d.h. der Benutzer erhält ein SYSLST-Protokoll, das ihm anzeigt, wie der IMPORT-Makroaufruf verarbeitet worden wäre. Das Protokoll enthält (abhängig von „pfadname“) eine Liste der Dateien auf dem mit VOLUME bezeichneten Datenträger und die Rückinformationen/Meldungen des IMPORT-Makroaufrufs.
Das DVS prüft zu diesem Zeitpunkt nicht, ob Dateisperren oder Schutzmerkmale evtl. einen Import verhindern. Der Benutzer muss also beim realen Import dafür sorgen, dass die Dateien nicht gesperrt sind und dass Schreibzugriff zugelassen ist.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben. In allen Makroaufrufen, die sich durch den MF-Operanden unterscheiden (MF=L/E/D/C), muss der Versionsoperand den gleichen Wert haben.

NUSERID = userid
Nur für die Systemverwaltung zulässig
Benutzerkennung, unter der die Datei zu katalogisieren ist. Die neue Userid wird ohne $ und ohne „.“ angegeben.
Einer Datei auf privater Platte wird sowohl im Dateikatalog als auch im F1-Kennsatz der Platte die neue Benutzerkennung zugewiesen.
Einer BS2000-Datei auf einem Net-Storage-Volume wird sowohl im Dateikatalog als auch im Katalog des Net-Storage-Volumes die neue Benutzerkennung zugewiesen.
Ein Node-File kann dagegen nicht unter der neuen Benutzerkennung katalogisiert werden, da der Eigentümer von Node-Files nicht geändert werden darf. Der Import wird in diesem Fall mit Returncode D zurückgewiesen.

pfadname
bezeichnet die Dateien, Dateigenerationsgruppen oder Dateigenerationen, die katalogisiert werden sollen, mit: <c-string 1..54: filename 1..54> (es ist auch ein teilqualifizierter Dateiname erlaubt).

Wird „pfadname“ nicht angegeben, katalogisiert das DVS alle Dateien usw., die unter der Benutzerkennung des laufenden Auftrags auf dem im VOLUME-Operanden angegebenen Datenträger gespeichert sind.

Pfadname bedeutet [$userid.]dateiname

userid
Benutzerkennung; der Anwender kann nur Dateien importieren, für die er das Eigentumsrecht hat.
Default-Userid: die Benutzerkennung des laufenden Auftrags (d.h. des Kommandos SET-LOGON-PARAMETERS bzw. LOGON).

dateiname
voll- oder teilqualifizierter Name einer Datei, Dateigenerationsgruppe oder Dateigeneration

Bei Dateigenerationen/Dateigenerationsgruppen zuerst den Gruppeneintrag erstellen, dann die Generationen katalogisieren!

PREFIX
Dieser Operand ist nur in Zusammenhang mit MF=D/C relevant.

Standardwert: I

= pre
Gibt den Zeichenvorrat für alle in der DSECT verwendeten Namen an.
Es ist ein Buchstabe zulässig.

= *
Gibt an, dass kein Zeichenvorsatz verwendet werden soll.

PVSID = catid
Gibt an, in welchem Pubset die Dateien katalogisiert werden sollen. Fehlt diese Angabe, werden die Katalogeinträge unter der Default-Catid der Benutzerkennung eingerichtet.

Wenn im Operanden VOLUME ein Net-Storage-Volume angegeben ist, dann muss diesem Net-Storage-Volume der Pubset zugeordnet sein, in dessen Katalog die Einträge importiert werden. Der Katalogeintrag im Pubset wird dann aktualisiert mit den Daten des Katalogeintrags des Net-Storage-Volumes.

REPLACE
Legt fest, ob ein bereits vorhandener „alter“ Katalogeintrag überschrieben werden soll.

= NO
Das DVS überschreibt den vorhandenen Katalogeintrag nicht.

= YES
Der alte Katalogeintrag wird gelöscht, wenn Diskrepanzen zu den Angaben im IMPORT-Makroaufruf bestehen.
- Die katalogisierte Datei ist auf gemeinschaftlicher Platte gespeichert: der Katalogeintrag wird überschrieben und die Public-Datei damit gelöscht (falls die Schutzmerkmale es zulassen und sie nicht gesperrt ist, sonst bleibt der alte Katalogeintrag erhalten).
- Die katalogisierte Datei steht auf Privatplatte, beginnt aber auf einer anderen Platte als im VOLUME-Operanden angegeben: der Katalogeintrag wird überschrieben (falls nicht durch Dateisperre oder Schutzmerkmale verhindert, sonst bleibt der alte Eintrag erhalten).

- Die katalogisierte Datei ist auf Net-Storage-Volume gespeichert. Eine Datei auf dem gleichen Net-Storage-Volume mit gleichem Namen wird nicht importiert und der Katalogeintrag wird nicht gelöscht.
- Die katalogisierte Datei ist auf Net-Storage-Volume gespeichert. Eine Datei auf Privatplatte, auf einem anderen Net-Storage-Volume oder auf dem gleichen Net-Storage-Volume jedoch mit unterschiedlichem Namen (kein Node-File) wird importiert. Der Katalogeintrag wird überschrieben und die Datei somit gelöscht (falls nicht durch Dateisperre oder Schutzmerkmale verhindert, sonst bleibt der alte Eintrag erhalten). Löschen der Datei bedeutet in diesem Fall:
  - Eine BS2000-Datei auf Net-Storage wird auch auf dem Net-Storage-Volume gelöscht.
  - Ein Node-File bleibt auf dem Net-Storage-Volume erhalten.

  Node-Files können im Gegensatz zu BS2000-Dateien nicht in eine andere Benutzerkennung importiert werden, da der Eigentümer von Node-Files nicht geändert werden darf.
- Die katalogisierte Datei steht auf Privatplatte und beginnt auf der im VOLUME-Operanden angegebenen Platte:
  Der Katalogeintrag wird *nicht* gelöscht (exportiert). Eine Datei gleichen Namens wird *nicht* importiert.

= ABS
Der alte Katalogeintrag wird überschrieben, auch wenn Katalogeintrag und Angaben im IMPORT-Makroaufruf übereinstimmen. Der Returncode zeigt an, ob der Eintrag überschrieben wurde (Returncode 8) oder ob Überschreiben wegen einer Dateisperre nicht möglich war (Returncode 9).

VERSION = 1
Steuert die Makrogenerierung. Es werden die Operandenliste und ggf. der SVC generiert, gültig für BS2000-Versionen ab V9.5.

Voreinstellung: es werden Operandenliste und SVC generiert wie in den BS2000-Versionen < V9.5

VOLUME = vsn
Gibt die Archivnummer („vsn“) des Datenträgers an, auf dem die zu importierenden Dateien gespeichert sind.

Hinweis zur Programmierung

Die Einzelinformationen sind jeweils 56 Byte lang und haben folgenden Aufbau:

pfadname	(54 Byte)	Returncode (2 Byte)

Returncodes

Für den Anwender ist nur das rechte Byte des Returncodes von Bedeutung, das ihm eine Zusatzinformation über die Verarbeitung des IMPORT-Makroaufrufs liefert, wenn der Returncode in Register 15 X'00' ist.

Returncode	Bedeutung
C'0'	die Datei wurde neu katalogisiert, eine Datei gleichen Namens existierte vorher nicht
C'1'	eine Datei gleichen Namens existierte bereits, sie wurde überschrieben; in Zusammenhang mit LIST=ONLY: eine Datei existiert bereits, Schutzmerkmale wurden nicht überprüft
C'2'	eine Datei gleichen Namens existiert, sie wurde nicht überschrieben; der Operand REPLACE hatte den Wert NO
C'3'	eine Datei gleichen Namens existiert bereits und konnte auf Grund aktiver Schutzfunktionen (ACCESS=READ, WRPASS usw.) nicht gelöscht werden oder: die Datei ist gesperrt, da sie gerade bearbeitet wird
C'4'	Systemfehler beim Zugriff auf den Katalog
C'5'	die Datei ist bereits katalogisiert und steht auf dem im VOLUME-Operanden angegebenen Datenträger
C'6'	Systemfehler beim Zugriff auf den F1-Kennsatz der Privatplatte bzw. den Katalog des Net-Storage-Volumes
C'7'	nicht erlaubter Import einer Dateigeneration: die absolute Generationsnummer der zu importierenden Generation ist nicht mit den im Gruppeneintrag festsetzten Grenzen verträglich
C'8'	der Katalogeintrag existierte bereits für die angegebene Platte; er wurde ersetzt
C'9'	der Katalogeintrag existiert bereits für die angegebene Platte, die Datei ist aber gesperrt
C'A'	der Pfadname der zu importierenden Datei (zusammmen mit catid und userid) ist länger als 54 Zeichen
C'B'	Fehler beim Zugriff auf den Net-Storage
C'C'	die zu importierende Datei ist größer 32 GB, aber der angegebene Pubset erlaubt keine Dateien größer 32 GB
C'D'	Das Importieren von Node-Files unter eine neue Benutzerkennung ist nicht erlaubt.

Eine Datei des Datenträgers wurde erfolgreich verarbeitet, wenn die Rückinformation C'0', C'1', C'5' oder C'8' ist.

In folgender Tabelle wird das linke Byte des Returncodes beschrieben, das jedoch nur dann von Bedeutung ist, wenn der Systemverwalter den Operanden NUSERID verwendet. Die Angaben beziehen sich auf Einträge im Systemkatalog unter „Ouserid“.

Returncode für System-verwalter	**Bedeutung**
C'0'	unter der Benutzerkennung, die im F1-Kennsatz einer privaten Platte bzw. dem Katalog des Net-Storage-Volumes steht, existiert kein Eintrag im Systemkatalog
C'1'	eine mit dem Namen bereits katalogisierte Datei wurde gelöscht. War der Operand LIST=ONLY angegeben, bedeutet dieser Wert lediglich, dass eine Datei gleichen Namens bereits existiert. Die Schutzmerkmale werden in diesem Fall nicht überprüft
C'2'	eine Datei gleichen Namens existiert bereits; der Operand REPLACE hatte den Wert NO
C'3'	die bereits katalogisierte Datei ist geschützt (Fehler beim Löschen dieser Datei)
C'4'	Systemfehler beim Lesen des Katalogs
C'D'	Das Importieren von Node-Files unter eine neue Benutzerkennung ist nicht erlaubt.

Verhalten bei Überlauf des Ausgabebereichs: Ein zu kleiner Benutzerbereich wird durch R15 = 05AB angezeigt. Es wird trotzdem importiert. Layout des Ausgabebereichs siehe oben; Endekriterium X'FF' auf Distanz: letzter Eintrag +X'36'

Der Returncode wird im Register 15 abgelegt. Bei ordnungsgemäßem Abschluss des Makroaufrufs wird der Inhalt des Registers 15 auf null gesetzt. Die möglichen Returncodes des DVS können durch den IDEMS-Makro erzeugt werden.

INSRT – Satz einfügen

Makrotyp: R bei PARMOD=24
0 bei PARMOD=31

Der INSRT-Makroaufruf überträgt einen Satz aus dem Anwenderbereich in eine Datei, und zwar an die Position, die durch den Satzschlüssel definiert ist.

Steht in der Datei bereits ein Satz mit dem angegebenen Schlüssel, wird der neue Satz nicht übertragen, gleichgültig, ob DUPEKY=YES definiert wurde oder nicht. Die Steuerung geht an den EXLST-Ausgang DUPEKY.
Das Einfügen eines Satzes mit bereits in der Datei vorhanden Schlüssel ist nur mit STORE möglich, sequenzielle Erweiterung mit PUT.

Bei RECFORM=V muss der Anwender vor Aufruf des INSRT-Makros das Satzlängenfeld mit der Länge des einzufügenden Satzes versorgen.

Format

Operation	Operanden
INSRT	$\left\{\begin{array}{l}\texttt{fcbadr}\\\texttt{(1)}\end{array}\right\},\left\{\begin{array}{l}\texttt{area}\\\texttt{(0)}\end{array}\right\}[,\texttt{PARMOD=}\left\{\begin{array}{l}\texttt{24}\\\texttt{31}\end{array}\right\}]$

Operandenbeschreibung

fcbadr
Adresse des FCB für die zu verarbeitende Datei

(1)
Die FCB-Adresse steht im Register 1.

area
Adresse des Satzes, der in die Datei eingefügt werden soll; auch im Ortungsbetrieb muss der Satz an der Adresse „area“ bereitgestellt werden.

(0)
Die Adresse des in die Datei einzufügenden Satzes steht im Register 0.

PARMOD
Gibt den Generierungsmodus an.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24
Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (nur 24-Bit-Adressierung).

= 31
Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist (24-Bit- oder 31-Bit-Adressierung).

Hinweis zur Programmierung

Der INSRT-Makroaufruf zerstört die Register 0, 1, 14 und 15.

ISREQ – Sperre aufheben

Makrotyp: R bei PARMOD=24
0 bei PARMOD=31

Der ISREQ-Makroaufruf wird bei Shared-Update-Verarbeitung genutzt, um eine Sperre aufzuheben, die explizit bei einer Leseoperation angefordert wurde und die nicht (implizit) durch Zurückschreiben des Satzes (oder einen anderen ISAM-Aktionsmakroaufruf) aufgehoben wird.

Sperren werden implizit aufgehoben durch den nächsten ISAM-Aktionsmakroaufruf, es sei denn es folgt ein OSTAT-Makroaufruf oder eine Leseoperation für eine andere Datei mit der Angabe NOLOCK.

Die Sperre kann sein

– eine Satzsperre (bei NK-ISAM, nichtsequenzielle Verarbeitung)
– eine Bereichssperre (bei NK-ISAM, sequenzielle Verarbeitung)
– eine Datenblocksperre (bei K-ISAM)

Format

Operation	Operanden
ISREQ	{ fcbadr | (1) },ACTION=UNLOCK[,PARMOD={ 24 | 31 }]

Operandenbeschreibung

fcbadr
Adresse des FCB der Datei, in der die Sperre besteht

(r)
Die FCB-Adresse steht im Register „r“.

ACTION = UNLOCK
Hebt eine externe Sperre auf.
Nach Ausführung von „ISREQ ...,UNLOCK“ kehrt ISAM zu der Anweisung zurück, die dem ISREQ-Makroaufruf folgt.

PARMOD
Gibt den Generierungsmodus an.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24
Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (nur 24-Bit-Adressierung).

= 31
Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist (24-Bit- oder 31-Bit-Adressierung).

Hinweis zur Programmierung

Der ISREQ-Makroaufruf zerstört die Register 0, 1, 14 und 15.

Returncode

Das FCB-Feld „ID1ECB“ enthält einen Returncode. Je nach Fehlerschlüssel ist auch das Register 1 davon betroffen:

ID1ECB	R1	Bedeutung
X' 0000'	unverändert	Sperre aufgehoben
X' 0A01'	verändert	für die Datei, deren FCB-Adresse in Register 1 hinterlegt ist, besteht eine Sperre für den Auftrag *)
X' 0A02'	unverändert	keine Sperre vorhanden
X' 0AA3'	unverändert	FCB ungültig

*) Der Benutzer kann die Sperre aufheben, indem er den ISREQ-Makro mit unverändertem Register 1 als erstem Operanden aufruft.

LBRET – Rückkehr aus Benutzer-Kennsatzroutine

Makrotyp: R-Typ

Der LBRET-Makroaufruf wird nur bei der Verarbeitung von Standard-Benutzerkennsätzen benötigt. Standard-Benutzerkennsätze sind:

UVL – Benutzer-Bandanfangskennsatz (User Volume Header Label; UVL)

UHL – Benutzer-Dateianfangskennsatz (User File Header Label; UHL)

UTL – Benutzer-Endekennsatz (User Trailer Label; UTL)

Für Nichtstandardkennsätze wird der EXRTN-Makroaufruf verwendet.

Format

Operation	Operanden
LBRET	{fcbadr / (1)}, {0 / 1 / 2 / (0)}[,PARMOD={24 / 31}]

Operandenbeschreibung

fcbadr
Adresse des FCB = Adresse des FCB-Makroaufrufs

(1)
die FCB-Adresse steht in Register 1

Der zweite Operand gibt einen Funktionsschlüssel an.

0
Die Kennsatzbehandlung wird beendet; bei Ausgabedateien wird auch der aktuelle Kennsatz nicht mehr geschrieben.

1
Die Kennsatzbehandlung wird beendet: bei Eingabedateien werden die weiteren Kennsätze der Gruppe übergangen; bei Ausgabedateien wird der aktuelle Kennsatz noch geschrieben.

2
Die Kennsatzbehandlung wird fortgesetzt: bei Eingabedateien folgen dem UVL- oder UTL-Kennsatz noch weitere Benutzerkennsätze, die vom Programm gelesen/verarbeitet werden sollen; bei Ausgabedateien gibt das System die Steuerung nach dem Schreiben eines Benutzerkennsatzes an das Benutzerprogramm zurück. Der Benutzer kann so maximal 9 UVL und 256 UHL- und UTL-Kennsätze schreiben. Das System beendet die Kennsatzverarbeitung, wenn diese Grenzen erreicht werden.

(0)
Das Register 0 enthält im rechten Byte den Funktionsschlüssel

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.

Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst. Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

Hinweis zur Programmierung

Der LBRET-Makroaufruf zerstört die Register 0, 1, 14 und 15.

LFFSNAP– Dateien von einem Snapset auflisten

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Mit dem Makro LFFSNAP kann sich der Benutzer über Dateien informieren, die bei der Pubset-Sicherung auf einen Snapset gesichert wurden. Die Informationen sind ausgerichtet auf die Restaurierbarkeit von Dateien (mit dem Makro RFFSNAP bzw. dem Kommando RESTORE-FILE-FROM-SNAPSET). Der zugehörige Pubset muss importiert sein.

Der nicht-privilegierte Benutzer kann sich über alle für ihn zugreifbaren Dateien informieren (wie bei dem FSTAT-Makro bzw. bei dem Kommando SHOW-FILE-ATTRIBUTES, die Informationen aus dem aktuellen Dateikatalog liefern).

Informationen über alle existierenden Snapsets zu einem Pubset können mit dem Kommando SHOW-SNAPSET-CONFIGURATION eingeholt werden.

Die Snapsets sind temporär nicht verfügbar, wenn das Subsystem SHC-OSD zum Zeitpunkt des Pubset-Imports noch nicht aktiv war. Der Makroaufruf wird in diesem Fall mit Returncode 0622 abgebrochen. Sobald SHC-OSD aktiv ist, werden die Snapsets bei Aufruf des Kommandos SHOW-SNAPSET-CONFIGURATION nachträglich aktiviert.

Privilegierte Funktionen

Die Systembetreuung (Privileg TSOS) kann sich über Dateien aller Benutzerkennungen informieren. Musterzeichen innerhalb der Benutzerkennung sind dabei nicht zulässig.

Format

Operation	Operanden
LFFSNAP	,PATHNAM=<c-string 1..80: filename 1..54 with-wild(80)> / <var: char:80> ,SNAPSET=<integer -52..-1> / *LATEST ,SNAPID=<c-string 1..1: name 1..1 with-low> / <var: char 1..1> ,OUTAREA=(<var: pointer>,<integer 0..32767>) ,EQUATES=*YES / *NO ,EXPAND=PARAM / OUTPUT
	MF=L
	MF=D,PREFIX=D / <pre>
	MF=E,PARAM=<name 1..27>
	MF=C / M ,PREFIX=D / <pre> ,MACID=MAL / <macid>

Operandenbeschreibung

PATHNAM
Auswahl der Dateien, die aufgelistet werden sollen.

=<c-string 1..80: filename 1..54 with-wild(80)>
Pfadname der Datei(en) auf dem Snapset. Mit Musterzeichen kann eine Auswahlangabe für eine Dateimenge erfolgen.

Es werden nur Dateien aufgelistet, die folgende Voraussetzungen erfüllen:
– Sie müssen zum Zeitpunkt der Snapset-Erstellung katalogisiert sein.
– Der Pubset, an dem sie katalogisiert sind, muss lokal importiert sein.
– Sie dürfen nicht auf Privatplatte liegen.

Die Angabe von Aliasnamen ist zulässig. Einzelne Dateigenerationen können angegeben werden. Bei Angabe einer Dateigenerationsgruppe werden die Dateigenerationen mit ausgegeben.

Der privilegierte Benutzer (Privileg TSOS) kann sich über Dateien aller Benutzerkennungen informieren. Dabei sind Musterzeichen in der Benutzerkennung nicht zulässig.

=<var: char:80>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 80 Byte, in dem der Pfadname bzw. die Musterzeichenfolge für die gewünschte(n) Datei(en) abgelegt ist.

SNAPSET
Der Operand darf nicht zusammen mit dem Operanden SNAPID angegeben werden.

Bezeichnet den Snapset, von dem die Datei-Informationen ausgegeben werden sollen, über das relative Alter.

=<integer -52..-1>
Bezeichnet den Snapset explizit über das relative Alter. Der Wert -1 entspricht dem jüngsten Snapset (entspricht auch *LATEST).

=*LATEST
Die Informationen werden von dem jüngsten Snapset (d.h. von der aktuellsten Pubset-Sicherung) ausgegeben.

SNAPID
Der Operand darf nicht zusammen mit dem Operanden SNAPSET angegeben werden.
Bezeichnet den Snapset, von dem die Datei-Informationen ausgegeben werden sollen.

=<c-string 1..1: name 1..1 with-low>
Bezeichnet den Snapset explizit über die Snapset-Id. Die maximal 52 Snapsets zu einem Pubset werden unterschieden durch Snapset-Ids aus den 26 Kleinbuchstaben a bis z und den 26 Großbuchstaben A bis Z.

=<var: char 1..1>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 1 Byte, in dem die Snapset-Id abgelegt ist.

Hinweis
Wenn weder SNAPSET noch SNAPID angegeben sind, wird die Information vom jüngsten Snapset ausgegeben.

OUTAREA
Bestimmt den Ausgabebereich, in dem die Informationen abgelegt werden sollen.

=(<var: pointer>,<integer 0..32767>)
Gibt Adresse und Länge des Ausgabebereichs an.

EQUATES
Steuerungs-Operand nur für MF=C und MF=D:
Gibt an, ob bei der Expansion des Parameter- oder Ausgabebereichs auch Equates für die Werte der Felder des Parameter- oder Ausgabebereichs generiert werden sollen.

= *YES
Bei der Expansion des Parameter- oder Ausgabebereichs werden auch Equates für die Werte der Felder des Parameter- oder Ausgabebereichs generiert.

= *NO
Bei der Expansion des Parameter- oder Ausgabebereichs werden keine Equates für die Werte der Felder des Parameter- oder Ausgabebereichs generiert.

XPAND

Steuerungs-Operand nur für MF=C und MF=D:

Es wird festgelegt, welche Struktur zu expandieren (erzeugen) ist. Angaben bei diesem Operanden werden bei anderen MF-Werten ignoriert.

= PARAM

Das Layout der Parameterliste wird expandiert.

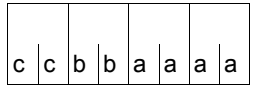
= OUTPUT

Das Layout des Ausgabebereiches wird expandiert.

Returncodes

Der Returncode wird im Standardheader des Parameterbereichs abgelegt. Der Parameterbereich darf dann nicht im Read-only-Bereich liegen, sonst erfolgt Programmterminierung.

Standard-header:



Über die Ausführung des Makros LFFSNAP wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2,bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'00'	X'0000'	Kein Fehler
X'00'	X'40'	X'0501'	angeforderter Katalog nicht verfügbar
X'00'	X'40'	X'0505'	Fehler bei der Rechner-Kommunikation
X'00'	X'40'	X'0512'	angeforderter Katalog nicht gefunden
X'00'	X'40'	X'051B'	gewünschte Benutzerkennung nicht im Pubset
X'00'	X'40'	X'051D'	LOGON-Passwort auf angegebenem Pubset anders
X'00'	X'20'	X'0531'	Unerwarteter Fehler beim Katalogzugriff
X'00'	X'40'	X'0535'	angegebene Datei nicht zugreifbar
X'00'	X'82'	X'0594'	nicht genügend virtueller Speicher
X'00'	X'01'	X'05AB'	Adresse des Ausgabebereichs falsch/nicht angegeben
X'02'	X'00'	X'05B6'	fehlerhafte Zeitkonvertierung GTIME-Makro
X'00'	X'20'	X'05C7'	interner Fehler im DVS
X'00'	X'40'	X'05FC'	angegebene Benutzerkennung nicht im Home-Pubset
X'00'	X'40'	X'0615'	Datei liegt auf einem nicht verfuegbaren Volume Set
X'00'	X'40'	X'0616'	Volume Set in SM-Pubset nicht zugreifbar
X'00'	X'40'	X'0622'	Snapset nicht verfügbar
X'00'	X'40'	X'0624'	Dateiname ungültig

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'40'	X'0684'	Datei existiert nicht
X'02'	X'00'	X'06CB'	Ausgabeinformation nicht vollständig übertragen
X'00'	X'01'	X'06CB'	Ausgabebereich zu klein
X'00'	X'40'	X'06CC'	kein Dateiname entspricht der angegebenen Musterzeichenfolge
X'00'	X'01'	X'06F7'	Ungültiger Operandenwert
X'00'	X'01'	X'06FD'	Parameterbereich ungültig oder nicht zugreifbar

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Layout der Operandenliste

Makroauflösung mit MF=D und EXPAND=PARAM, sowie Standardwerten für EQUATES, PREFIX und MACID:

```
         LFFSNAP MF=D,XPAND=PARAM
DMALLFPL DSECT ,
DMALHDR  DS    0A
DMALFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
DMALIFID DS    0A              0   INTERFACE IDENTIFIER
DMALFCTU DS    AL2             0   FUNCTION UNIT NUMBER
DMALFCT  DS    AL1             2   FUNCTION NUMBER
DMALFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
DMALRET  DS    0A              4   GENERAL RETURN CODE
DMALSRET DS    0AL2            4   SUB RETURN CODE
DMALSR2  DS    AL1             4   SUB RETURN CODE 2
DMALSR1  DS    AL1             5   SUB RETURN CODE 1
DMALMRET DS    0AL2            6   MAIN RETURN CODE
DMALMR2  DS    AL1             6   MAIN RETURN CODE 2
DMALMR1  DS    AL1             7   MAIN RETURN CODE 1
DMALFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
DMALPNAM DS    CL80                    PATHNAM
DMALSNAP DS    FL1                     SNAPIND
*   SNAPSET - VALUES
DMALSNIN EQU   0                       SNAPSET=<integer>
DMALSNCH EQU   1                       SNAPSET=<char>
DMALSNLT EQU   2                       SNAPSET=*LATEST
*
DMALSNID DS    CL1                     SNAPID
DMALSNVL DS    H                       SNAPVALUE
DMALARAD DS    A                       OUTAREA=(<addr>,...)
DMALARLN DS    F                       OUTAREA=(...,<length>)
DMAL#    EQU   *-DMALHDR
```

Format des Ausgabebereiches

Makroauflösung mit MF=D und EXPAND=OUTPUT, sowie mit Standardwerten für EQUATES, PREFIX und MACID:

```
         LFFSNAP MF=D,XPAND=OUTPUT
         MFTST MF=D,PREFIX=D,MACID=MAL,ALIGN=F,

               DMACID=MAL,SUPPORT=(E,D,C,M,L),DNAME=MALOUTL
DMALOUTL DSECT ,
*,##### PREFIX=D, MACID=MAL #####
*   Snapset Output
DMALFSIZ DS    F                       FILESIZE
```

```
DMALOPNM DS    CL54                    PATHNAME
DMALSTATE DS    FL1                     STATE
*   STATE = VALUES
DMALSTOP EQU   0                       STATE = OPENED
DMALSTCL EQU   1                       STATE = CLOSED
DMALSTNR EQU   2                       STATE = NOREST
*
DMALFTYPE DS    FL1                     FILETYPE
*   FTYPE = VALUES
DMALFTPB EQU   0                       FTYPE = PUBLIC
DMALFTMG EQU   1                       FTYPE = MIGRATED
DMALFTFG EQU   2                       FTYPE = FGG
DMALFTWR EQU   3                       FTYPE = WORK
DMALFTPD EQU   4                       FTYPE = PRDISK
DMALFTTP EQU   5                       FTYPE = TAPE
DMALFTNT EQU   6                       FTYPE = NET
*
*
DMALCRDT DS    0XL16                   Creation Date
DMALCRYE DS    CL4                     YEAR
DMALCRMO DS    CL2                     MONTH
DMALCRDA DS    CL2                     DAY
DMALCRHO DS    CL2                     HOURS
DMALCRMI DS    CL2                     MINUTES
DMALCRSE DS    CL2                     SECONDS
DMALCRUS DS    CL2                     UNUSED
*
*
DMALLCDT DS    0XL16                   Last Change Date
DMALLCYE DS    CL4                     YEAR
DMALLCMO DS    CL2                     MONTH
DMALLCDA DS  CL2             DAY
DMALLCHO DS    CL2                     HOURS
DMALLCMI DS    CL2                     MINUTES
DMALLCSE DS    CL2                     SECONDS
DMALLCUS DS    CL2                     UNUSED
*
DMALENLT DS    FL1                     END Indicator
*
DMALSNXT EQU   0                       FURTHER ENTRY
DMALSNED EQU   1                       LAST ENTRY
DMALSNNS EQU   2                       NOT ENOUGH SPACE
*
DMALUNUS DS    XL3                     UNUSED
DMALOUTPUT# EQU  *-DMALFSIZ
```

Bei Ausgabe der Snapset-Information in den Ausgabebereich des Anwenders werden folgende Fälle unterschieden:

– Die Ausgabe war vollständig möglich
  Der Ausgabebereich wird mit der gewünschten Information überschrieben, der Aufrufer erhält den Returncode 0. Der Ausgabebereich wird nicht bis zum Ende durchgelöscht, sondern nur so weit beschrieben wie erforderlich.

– Keine Dateien entsprechen den Auswahlkriterien
  Der Ausgabebereich wird überhaupt nicht beschrieben. Der Aufrufer erhält den Returncode 0684 oder 06CC (bei Musterzeichen/Teilqualifizierung).

– Die Ausgabe war nicht möglich
  Der Ausgabebereich konnte nicht beschrieben werden (Returncode 05AB nach Validierung des Ausgabebereichs bzw. der Adresse) oder er ist zu klein um eine Ausgabeinformation zu übertragen (Returncode 06CB).

– Die Ausgabe war nicht vollständig möglich
  Einige Datei-Informationsblöcke konnten nicht übertragen werden. Zusätzlich zur entsprechenden Anzeige im Ausgabebebereich (NOT ENOUGH SPACE) wird Returncode 06CB mit Subreturncode2 X'02' ausgegeben.

Beispiel für eine Aufruffolge

```
         LFFSNAP MF=D,XPAND=OUTPUT
              .
              .
         MVC   LFFSMFC(DMAL#),LFFSMFL
         LFFSNAP MF=M,PATHNAM=':X:T.1',PARAM=LFFSMFC,PREFIX=X,         *
               SNAPSET=-1,OUTAREA=(AREAAD,100)
         LFFSNAP MF=E,PARAM=LFFSMFC
              .
              .
LFFSMFC  LFFSNAP MF=C,PREFIX=X,XPAND=PARAM
LFFSMFL  LFFSNAP MF=L,PATHNAM='X'
AREA     DS    CL100
AREAAD   DC    A(AREA)
             .
             .
```

LJFSNAP– Jobvariablen von einem Snapset auflisten

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Mit dem Makro LJFSNAP kann sich der Benutzer über Jobvariablen informieren, die bei der Pubset-Sicherung auf einen Snapset gesichert wurden. Die Informationen sind ausgerichtet auf die Restaurierbarkeit von Jobvariablen (mit dem Makro RJFSNAP bzw. dem Kommando RESTORE-JV-FROM-SNAPSET). Der zugehörige Pubset muss importiert sein.

Der nicht-privilegierte Benutzer kann sich über alle für ihn zugreifbaren Jobvariablen informieren (wie bei SHOW-JV-ATTRIBUTES, das Informationen aus dem aktuellen Dateikatalog liefert).

Informationen über alle existierenden Snapsets zu einem Pubset können mit dem Kommando SHOW-SNAPSET-CONFIGURATION eingeholt werden.

Die Snapsets sind temporär nicht verfügbar, wenn das Subsystem SHC-OSD zum Zeitpunkt des Pubset-Imports noch nicht aktiv war. Der Makroaufruf wird in diesem Fall mit Returncode 0622 abgebrochen. Sobald SHC-OSD aktiv ist, werden die Snapsets bei Aufruf des Kommandos SHOW-SNAPSET-CONFIGURATION nachträglich aktiviert.

Privilegierte Funktionen

Die Systembetreuung (Privileg TSOS) kann sich über Jobvariablen aller Benutzerkennungen informieren. Musterzeichen innerhalb der Benutzerkennung sind dabei nicht zulässig.

Format

Operation	Operanden
LJFSNAP	,JVNAME=<c-string 1..80: filename 1..54 with-wild(80)> / <var: char:80> ,SNAPSET=<integer -52..-1> / *LATEST ,SNAPID=<c-string 1..1: name 1..1 with-low> / <var: char 1..1> ,OUTAREA=(<var: pointer>,<integer 0..32767>) ,EQUATES=*YES / *NO ,EXPAND=PARAM / OUTPUT
	MF=L
	MF=D,PREFIX=D / <pre>
	MF=E,PARAM=<name 1..27>
	MF=C / M ,PREFIX=D / <pre> ,MACID=MAJ / <macid>

Operandenbeschreibung

JVNAME
Auswahl der Jobvariablen, die aufgelistet werden sollen.

=<c-string 1..80: filename 1..54 with-wild(80)>
Pfadname der Jobvariablen auf dem Snapset. Mit Musterzeichen kann eine Auswahlangabe für eine Jobvariablenmenge erfolgen.

Die Jobvariablen müssen folgende Voraussetzungen erfüllen:
– Sie müssen zum Zeitpunkt der Snapset-Erstellung katalogisiert sein.
– Der Pubset, an dem sie katalogisiert sind, muss lokal importiert sein.

Die Angabe von Aliasnamen ist zulässig.

Der privilegierte Benutzer (Privileg TSOS) kann sich über Jobvariablen aller Benutzerkennungen informieren. Dabei sind Musterzeichen in der Benutzerkennung nicht zulässig.

=<var: char:80>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 80 Byte, in dem der Pfadname bzw. die Musterzeichenfolge für die gewünschte(n) Jobvariable(n) abgelegt ist.

SNAPSET
Der Operand darf nicht zusammen mit dem Operanden SNAPID angegeben werden.
Bezeichnet den Snapset, von dem die Jobvariablen-Informationen ausgegeben werden sollen, über das relative Alter.

=<integer -52..-1>
Bezeichnet den Snapset explizit über das relative Alter. Der Wert -1 entspricht dem jüngsten Snapset (entspricht auch *LATEST).

=*LATEST
Die Informationen werden von dem jüngsten Snapset (d.h. von der aktuellsten Pubset-Sicherung) ausgegeben.

SNAPID
Der Operand darf nicht zusammen mit dem Operanden SNAPSET angegeben werden.
Bezeichnet den Snapset, von dem restauriert werden soll, über die Snapset-Id.

=<c-string 1..1: name 1..1 with-low>
Bezeichnet den Snapset explizit über die Snapset-Id. Die maximal 52 Snapsets zu einem Pubset werden unterschieden durch Snapset-Ids aus den 26 Kleinbuchstaben a bis z und den 26 Großbuchstaben A bis Z.

=<var: char 1..1>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 1 Byte, in dem die Snapset-Id abgelegt ist.

Hinweis
Wenn weder SNAPSET noch SNAPID angegeben sind, wird die Information vom jüngsten Snapset ausgegeben.

OUTAREA
Bestimmt den Ausgabebereich, in dem die Informationen abgelegt werden sollen.

=(<var: pointer>,<integer 0..32767>)
Gibt Adresse und Länge des Ausgabebereichs an.

EQUATES

Steuerungs-Operand nur für MF=C und MF=D:

Gibt an, ob bei der Expansion des Parameter- oder Ausgabebereichs auch Equates für die Werte der Felder des Parameter- oder Ausgabebereichs generiert werden sollen.

= *YES

Bei der Expansion des Parameter- oder Ausgabebereichs werden auch Equates für die Werte der Felder des Parameter- oder Ausgabebereichs generiert.

= *NO

Bei der Expansion des Parameter- oder Ausgabebereichs werden keine Equates für die Werte der Felder des Parameter- oder Ausgabebereichs generiert.

XPAND

Steuerungs-Operand nur für MF=C und MF=D:

Es wird festgelegt, welche Struktur zu expandieren (erzeugen) ist. Angaben bei diesem Operanden werden bei anderen MF-Werten ignoriert.

= PARAM

Das Layout der Parameterliste wird expandiert.

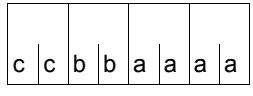
= OUTPUT

Das Layout des Ausgabebereiches wird expandiert.

Returncodes

Der Returncode wird im Standardheader des Parameterbereichs abgelegt. Der Parameterbereich darf dann nicht im Read-only-Bereich liegen, sonst erfolgt Programmterminierung.

Standard-header:



Über die Ausführung des Makros LJFSNAP wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2,bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'00'	X'0000'	kein Fehler
X'00'	X'40'	X'0501'	angeforderter Katalog nicht verfügbar
X'00'	X'40'	X'0505'	Fehler bei der Rechner-Kommunikation
X'00'	X'40'	X'0512'	angeforderter Katalog nicht gefunden
X'00'	X'40'	X'051B'	gewünschte Benutzerkennung nicht im Pubset
X'00'	X'40'	X'051D'	LOGON-Passwort auf angegebenem Pubset anders
X'00'	X'20'	X'0531'	Unerwarteter Fehler beim Katalogzugriff

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'82'	X'0594'	nicht genügend virtueller Speicher
X'00'	X'01'	X'05AB'	Adresse des Ausgabebereichs falsch/nicht angegeben
X'02'	X'00'	X'05B6'	fehlerhafte Zeitkonvertierung GTIME-Makro
X'00'	X'20'	X'05C7'	interner Fehler im DVS
X'00'	X'40'	X'05FC'	angegebene Benutzerkennung nicht im Home-Pubset
X'00'	X'40'	X'0622'	Snapset nicht verfügbar
X'00'	X'40'	X'0624'	JV-Name ungültig
X'00'	X'40'	X'0682'	JV Fehler beim Zugriff auf JV
X'02'	X'00'	X'06CB'	Ausgabeinformation nicht vollständig übertragen
X'00'	X'01'	X'06CB'	Ausgabebereich zu klein
X'00'	X'01'	X'06F7'	Ungültiger Operandenwert
X'00'	X'01'	X'06FD'	Parameterbereich ungültig oder nicht zugreifbar

Die Returncodes mit dem Maincode X’04xy’ gehören zur Komponente JVS. Eine Liste mit den Erläuterungen kann mit dem Makro JVSERROR ausgegeben werden (siehe auch Handbuch „Jobvariablen“ [21]).

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Layout der Operandenliste

Makroauflösung mit MF=D und EXPAND=PARAM, sowie Standardwerten für EQUATES, PREFIX und MACID:

```
         LJFSNAP MF=D,XPAND=PARAM
DMAJLFPL DSECT ,
DMAJHDR  DS    0A
DMAJFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
DMAJIFID DS    0A              0   INTERFACE IDENTIFIER
DMAJFCTU DS    AL2             0   FUNCTION UNIT NUMBER
DMAJFCT  DS    AL1             2   FUNCTION NUMBER
DMAJFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
DMAJRET  DS    0A              4   GENERAL RETURN CODE
DMAJSRET DS    0AL2            4   SUB RETURN CODE
DMAJSR2  DS    AL1             4   SUB RETURN CODE 2
DMAJSR1  DS    AL1             5   SUB RETURN CODE 1
DMAJMRET DS    0AL2            6   MAIN RETURN CODE
DMAJMR2  DS    AL1             6   MAIN RETURN CODE 2
DMAJMR1  DS    AL1             7   MAIN RETURN CODE 1
DMAJFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
DMAJJVNM DS    CL80                    JVname
DMAJSNAP DS    FL1                     Snapind
*   SNAPSET - VALUES
DMAJSNIN EQU   0                       SNAPSET=<integer>
DMAJSNCH EQU   1                       SNAPSET=<char>
DMAJSNLT EQU   2                       SNAPSET=*LATEST
*
DMAJSNID DS    CL1                     Snapid
DMAJSNVL DS    H                       SnapValue
DMAJARAD DS    A                       Outarea=(<addr>,...)
DMAJARLN DS    F                       Outarea=(...,<length>)
DMAJ#    EQU   *-DMAJHDR
```

Format des Ausgabebereiches

Makroauflösung mit MF=D und EXPAND=OUTPUT, sowie mit Standardwerten für EQUATES, PREFIX und MACID:

```
         LJFSNAP MF=D,XPAND=OUTPUT
DMAJOUTL DSECT ,
*   Output List
DMAJJSIZ DS    F                       JVSIZE
DMAJOJVN DS    CL54                    JVNAME
DMAJUNU1 DS    XL2                     UNUSED
*
DMAJCRDT DS    0XL16                   Creation Date
DMAJCRYE DS    CL4                     YEAR
DMAJCRMO DS    CL2                     MONTH
DMAJCRDA DS    CL2                     DAY
DMAJCRHO DS    CL2                     HOURS
DMAJCRMI DS    CL2                     MINUTES
DMAJCRSE DS    CL2                     SECONDS
DMAJCRUS DS    CL2                     UNUSED
*
*
DMAJEXDT DS    0XL16                   Expiration Date
DMAJEXYE DS    CL4                     YEAR
DMAJEXMO DS    CL2                     MONTH
DMAJEXDA DS    CL2                     DAY
DMAJEXHO DS    CL2                     HOURS
DMAJEXMI DS    CL2                     MINUTES
DMAJEXSE DS    CL2                     SECONDS
DMAJEXUS DS    CL2                     UNUSED
*
DMAJENLT DS    FL1                     END Indicator
*
DMAJSNXT EQU   0                       FURTHER ENTRY
DMAJSNED EQU   1                       LAST ENTRY
DMAJSNNS EQU   2                       NOT ENOUGH SPACE
*
DMAJUNU2 DS    XL3                     UNUSED
DMAJOUTPUT# EQU  *-DMAJJSIZ
```

Bei Ausgabe der Snapset-Information in den Ausgabebereich des Anwenders werden folgende Fälle unterschieden:

– Die Ausgabe war vollständig möglich
  Der Ausgabebereich wird mit der gewünschten Information überschrieben, der Aufrufer erhält den Returncode 0. Der Ausgabebereich wird nicht bis zum Ende durchgelöscht, sondern nur so weit beschrieben wie erforderlich.

– Keine Dateien entsprechen den Auswahlkriterien
  Der Ausgabebereich wird überhaupt nicht beschrieben. Der Aufrufer erhält den Returncode 0684 oder 06CC (bei Musterzeichen/Teilqualifizierung).
– Die Ausgabe war nicht möglich
  Der Ausgabebereich konnte nicht beschrieben werden (Returncode 05AB nach Validierung des Ausgabebereichs bzw. der Adresse) oder er ist zu klein um eine Ausgabeinformation zu übertragen (Returncode 06CB).
– Die Ausgabe war nicht vollständig möglich
  Einige Datei-Informationsblöcke konnten nicht übertragen werden. Zusätzlich zur entsprechenden Anzeige im Ausgabebebereich (NOT ENOUGH SPACE) wird Returncode 06CB mit Subreturncode2 X'02' ausgegeben.

Beispiel für eine Aufruffolge

```
       LJFSNAP MF=D,XPAND=OUTPUT
            .
            .
       MVC   LJFSMFC(DMAL#),LJFSMFL
       LJFSNAP MF=M,PREFIX=X,JVNAME=':X:JV.1',OUTAREA=(AREAAD,100), *
             PARAM=LJFSMFC
       LJFSNAP MF=E,PARAM=LJFSMFC
            .
            .
LJFSMFC  LJFSNAP MF=C,PREFIX=X,XPAND=PARAM
LJFSMFL  LJFSNAP MF=L,PATHNAM='X'
AREA     DS    CL100
AREAAD   DC    A(AREA)
```

MAILFIL– Datei per E-Mail versenden

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Der Makro MAILFIL sendet, analog zum Kommando MAIL-FILE, eine Datei als Anhang einer E-Mail. Als Empfänger der E-Mail wird eine Benutzerkennung angegeben. Absender ist die Benutzerkennung der aufrufenden Task. Aus dem EMAIL-ADDRESS-Feld dieser Benutzereinträge übernimmt MAIL-FILE die dort eingetragene E-Mail-Adresse. Die Ermittlung der Empfänger- und Absender-Adresse, insbesondere im Falle einer Adressliste, ist im Abschnitt „Selektion von E-Mail-Adressen über den Jobnamen“ auf Seite 748 beschrieben.

Versendet werden kann ein PLAM-Bibliothekselement, eine SAM- oder ISAM-Datei sowie der Inhalt der Systemdatei SYSLST bzw. SYSOUT. Eine PAM-Datei kann nur versendet werden, wenn der Inhalt im PDF-Format vorliegt.
Die Benutzertask, unter der MAILFIL ausgeführt wird, muss die nötigen Zugriffsrechte besitzen. Bei der automatischen Zeichensatz-Konvertierung wird das Dateiattribut CCS-Name ausgewertet.
Optional kann der Aufrufer vereinbaren, dass die Datei nach dem Versenden automatisch gelöscht werden soll.

Zur Makroausführung muss die Funktion „Mail-Sender“ des Software-Produkts interNet-Services zur Verfügung stehen und im Benutzereintrag der Systemkennung TSOS muss mindestens eine E-Mail-Adresse eingetragen sein.

Der Aufruf wird abgewiesen, wenn im Benutzereintrag des Empfängers keine E-Mail-Adresse eingetragen ist. Wenn für den Aufrufer keine E-Mail-Adresse eingetragen ist. wird als Absender ersatzweise die Adresse des Empfängers oder TSOS eingesetzt.

Falls die E-Mail-Versand nicht zugestellt werden kann (z.B. wegen ungültiger Adresse), wird eine Bounce-Mail an die E-Mail-Adresse von TSOS gesendet um die Systembetreuung zur Überprüfung der fehlerhaften Adresse aufzufordern. Wenn für TSOS mehrere E-Mail-Adressen eingetragen sind, wird für die Bounce-Mail die erste Adresse verwendet.

Die MAIL-FILE-Funktionalität wird auch von anderen Komponenten des BS2000 zum Versenden von Protokolldateien verwendet:

– bei der Auftragsbeendigung
  In den Kommandos EXIT-JOB (bzw. LOGOFF), CANCEL-JOB und ENTER-PROCEDURE kann statt der Druckausgabe auch das Versenden von SYSLST bzw. SYSOUT bei Auftragsbeendigung angefordert werden. Mit der Voreinstellung *STDOUT erfolgt die Ausgabe über das im Systemparameter SSMOUT festgelegte Ausgabemedium (Drucker oder E-Mail).
– bei Ausgaben von Dienstprogrammen
  Derzeit unterstützen HSMS ab V9.0 und MAREN ab V12.0 das Versenden von Ausgabeinformationen bzw. Protokollen.

Format

Operation	Operanden
MAILFIL	,PATHNAM=<c-string 1..54: filename 1..54> / <var: char:54> / *SYSOUT / *SYSLST / (*SYSLST,<integer 1..99>) ,LIBELEM=*NONE / (<c-string 1..64: composed-name 1..64> with under / <var: char:64> ,*HIGHEST / *UPPER / <c-string 1..24: composed-name 1..24> with under / <var: char:24> ,<c-string 1..8: alphanum-name 1..8> / <var: char:8>) ,USERID=*OWN / <c-string 1..8: name 1..8> / <var: char:8> ,SUBJECT=*STD / <c-string 1..256 with low> / <var: char:256> ,DELETE=*NO / *YES / *DESTROY ,EQUATES=*YES / *NO ,VERSION=1 / <integer 1..2> MF=L
	MF=D,PREFIX=D / <pre>
	MF=E,PARAM=<name 1..27>
	MF=C / M ,PREFIX=D / <pre> ,MACID=MAM / <macid>

Operandenbeschreibung

PATHNAM
Wählt die zu versendende Datei bzw. die PLAM-Bibliothek des zu versendenden Bibliothekselements aus.

=<c-string 1..54: filename 1..54>
Pfadname bzw. Bibliotheksname.
Für zu versendende Dateien gelten folgende Einschränkungen:
- Die Datei ist eine SAM- oder ISAM-Datei. Eine PAM-Datei wird nur versandt, wenn der Inhalt PDF-Format hat.
- Die Datei darf nicht leer sein.
- Der Name darf eine einzelne Dateigeneration bezeichnen, nicht jedoch eine Dateigenerationsgruppe.
- Es kann auch eine temporäre Datei sein.
  Ein privilegierter Benutzer (Privileg TSOS) kann auch temporäre Dateien einer anderen Task angeben. Er kann auch temporäre Dateien angeben, die auf einem anderen Pubset als dem Default-Pubset der Benutzerkennung liegen.
- Es darf keine Datei sein, die nur über eine RFA-Verbindung erreichbar ist.

=<var: char:54>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 54 Byte, in dem der Pfadname der zu sendenden Datei abgelegt ist.

=*SYSOUT
Bezeichnet die Systemdatei SYSOUT. Die Angabe ist nur möglich, wenn der SYSOUT-Datei eine Datei bzw. Dateigeneration auf Platte zugewiesen ist, die mit der Zugriffsmethode SAM erstellt wurde.
Die Angabe wird in folgenden Fällen abgewiesen:
– Die zugewiesene Datei ist noch leer.
– Die SYSOUT-Datei hat die Primärzuweisung.
– Es ist die Pseudodatei DUMMY, eine temporäre Datei, ein PLAM-Bibliothekselement oder eine S-Variable zugewiesen.

=*SYSLST
Bezeichnet die Systemdatei SYSLST. In folgenden Fällen wird die Angabe abgewiesen:
– SYSLST ist leer.
– Es ist die Pseudodatei DUMMY, eine temporäre Datei, ein PLAM-Bibliothekselement oder eine S-Variable zugewiesen.
– Die zugewiesene Datei bzw. Dateigeneration liegt nicht auf Platte oder wurde nicht mit der Zugriffsmethode SAM erstellt.

=(*SYSLST,<integer 1..99>)
Bezeichnet eine SYSLST-Datei aus der Menge SYSLST01 bis SYSLST99. Die Angabe ist nur möglich, wenn der SYSLST-Datei eine Datei bzw. Dateigeneration auf Platte zugewiesen ist, die mit der Zugriffsmethode SAM erstellt wurde.
Die Angabe wird in folgenden Fällen abgewiesen:
– Die zugewiesene Datei ist noch leer.
– Die SYSLST-Datei hat die Primärzuweisung.
– Es ist die Pseudodatei DUMMY, eine temporäre Datei, ein PLAM-Bibliothekselement oder eine S-Variable zugewiesen.

LIBELEM
Zu sendendes PLAM-Bibliothekselement. Zusätzlich muss im Operanden PATHNAM der entsprechende Name der PLAM-Bibliothek angegeben werden.
Es können nur Textelemente und PDF-Dateien versandt werden. Textelemente sind Elemente der Typen S, M, J, P, D, X und davon abgeleiteter Typen, soweit sie keine blockorientierten Sätze enthalten. Ein Element, das blockorientierte Sätze enthält, wird nur versandt, wenn sein Inhalt PDF-Format hat.

=*NONE
Es soll kein Bibliothekselement versendet werden. Der Operand PATHNAM enthält die Angaben über die zu versendenden Datei.

=(<element>,<version>,<typ>)
Gibt Name, Versionsnummer und Typ eines zu versendenden Elementes der im Operanden PATHNAM angegebenen PLAM-Bibliothek an.

<element>=<c-string 1..64: composed-name 1..64> with under
Name des zu versendenden Bibliothekselementes.

<element>=<var: char:64>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 64 Byte, in dem der Name des zu versendenden Bibliothekselementes abgelegt ist.

<version>=*HIGHEST
Wählt die höchste existierende Version aller Elemente aus, die den angegebenen Namen und den angegebenen Typ haben.

<version>=*UPPER
Wählt die höchste mögliche Version (X'FF') aller Elemente aus, die den angegebenen Namen und den angegebenen Typ haben.

<version>=<c-string 1..24: composed-name 1..24> with under
Version des zu versendenden Bibliothekselementes.

<version>=<var: char:24>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 24 Byte, in dem die Version des zu versendenden Bibliothekselementes abgelegt ist.

<typ>=<c-string 1..8: alphanum-name 1..8>
Typ des zu versendenden Bibliothekselementes.

<version>=<var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 8 Byte, in dem der Typ des zu versendenden Bibliothekselementes abgelegt ist.

USERID
Benutzerkennung, deren Eintrag im Benutzerkatalog die E-Mail-Adresse des Empfängers enthält.

=*OWN
Voreingestellt ist *OWN, d.h. die Logon-Benutzerkennung der aufrufenden Task. Wenn der Benutzereintrag eine Liste mit mehrerern E-Mail-Adressen enthält, wird ggf. eine Empfängeradresse in Abhängigkeit vom Jobnamen ausgewählt (siehe „Selektion von E-Mail-Adressen über den Jobnamen“ auf Seite 748).

=<c-string 1..8: name 1..8>
Benutzerkennung, aus deren Benutzereintrag die E-Mail-Adresse des Empfängers ermittelt wird.

=<var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 8 Byte, in dem die Benutzerkennung des Empfängers abgelegt ist.

SUBJECT
Bezeichnet den Betreff der E-Mail.

Hinweis
Da der BCAM-Name des absendenden BS2000-Systems bereits im Text der gesendeten E-Mail enthalten ist, muss er nicht extra in den Betreff aufgenommen werden.

=*STD
Mit *STD erhält die E-Mail einen standardisierten Betreff-Text, der neben dem Hinweis „von BS2000“ auch die Absenderkennung und den Dateinamen enthält.

=<c-string 1..256 with-low>
Betreff der E-Mail. Groß-/Kleinschreibung werden unterschieden. Es wird empfohlen, sich auf den internationalen Zeichensatz zu beschränken.

=<var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 256 Byte, in dem der Betreff der E-Mail abgelegt ist.

DELETE
Gibt an, ob die Datei oder das PLAM-Bibliothekselement nach dem erfolgreichen Versenden automatisch gelöscht werden soll. Dabei ist zu beachten:
– Wenn die Systemdatei SYSLST zu senden ist und SYSLST die Primärzuweisung besitzt, gilt DELETE=*YES.
– Wenn die Systemdatei SYSLST bzw. SYSOUT zu senden ist und die Systemdatei einer Datei bzw. Dateigeneration zugewiesen ist, unterbleibt das automatische Löschen.

=*NO
Die Datei oder das PLAM-Bibliothekselement wird nicht gelöscht. Die Datei oder das PLAM-Bibliothekselement ist nach dem Aufruf von MAIL-FILE sofort wieder verfügbar.

=*YES
Die Datei oder das PLAM-Bibliothekselement wird nach erfolgreichem Senden automatisch gelöscht. Die Datei oder das PLAM-Bibliothekselement gilt auch dann als erfolgreich gesendet, wenn sie danach nicht zugestellt werden kann (z.B. wegen unbekannter E-Mail-Adresse).

=*DESTROY
Die Angabe wirkt wie DELETE=*YES. Zusätzlich wird der Datei- oder Elementinhalt beim Löschen mit binär null überschrieben.

EQUATES
Steuerungs-Operand nur für MF=C und MF=D:
Gibt an, ob bei der Expansion des Parameterbereichs auch Equates für die Werte der Felder des Parameterbereichs generiert werden sollen.

= *YES
Bei der Expansion des Parameterbereichs werden auch Equates für die Werte der Felder des Parameterbereichs generiert.

= *NO
Bei der Expansion des Parameterbereichs werden keine Equates für die Werte der Felder des Parameterbereichs generiert.

VERSION
Steuert die Generierung des Parameterbereichs bzw. des Funktionsaufrufs. Der Operand muss bei der Generierung des Funktionsaufrufs denselben Wert haben wie bei der Generierung des zugehörigen Parameterbereichs.

= 1
erzeugt den für BS2000/OSD-BC V8.0 gültigen Parameterbereich bzw. Funktionsaufruf (die Angabe eines Bibliothekselements ist nicht möglich).

= 2
erzeugt den ab BS2000/OSD-BC V9.0 gültigen Parameterbereich bzw. Funktionsaufruf.

Layout des Parameterbereichs

Der Parameterbereich muss auf Wortgrenze ausgerichtet sein. Er beginnt mit einem Standardheader, den MAILFIL wie folgt initialisiert:

Function Unit Number	22
Function Number	32
Interface Version Number	2 bei Angabe von VERSION=2, sonst 1
Returncode	-1

Makroauflösung mit MF=D und Standardwerten für EQUATES, PREFIX und MACID:

```
         MAILFIL MF=D,VERSION=2
DMAMDEPL DSECT ,
DMAMHDR  DS    0A
DMAMFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
DMAMIFID DS    0A              0   INTERFACE IDENTIFIER
DMAMFCTU DS    AL2             0   FUNCTION UNIT NUMBER
DMAMFCT  DS    AL1             2   FUNCTION NUMBER
DMAMFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
DMAMRET  DS    0A              4   GENERAL RETURN CODE
DMAMSRET DS    0AL2            4   SUB RETURN CODE
DMAMSR2  DS    AL1             4   SUB RETURN CODE 2
DMAMSR1  DS    AL1             5   SUB RETURN CODE 1
DMAMMRET DS    0AL2            6   MAIN RETURN CODE
DMAMMR2  DS    AL1             6   MAIN RETURN CODE 2
DMAMMR1  DS    AL1             7   MAIN RETURN CODE 1
DMAMFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
DMAMPNAM DS    CL54                    Dateiname
DMAMUSID DS    CL8                     Benutzerkennung oder Blank
DMAMSUBJ DS    CL256                   Betreff oder Blank
DMAMDELE DS    FL1                     automatisches Loeschen
*   DELETE - values
DMAMDELN EQU   0                       DELETE = NO
DMAMDELY EQU   1                       DELETE = YES
DMAMDELD EQU   2                       DELETE = DESTROY
*
DMAMPNSP DS    FL1                     Typ der PATHNAM-Angabe
*   PATHNAM - values
DMAMBYFN EQU   0                       PATHNAM = fnam
DMAMSLST EQU   1                       PATHNAM = *SYSLST
```

```
DMAMBYSN EQU   2                          (*SYSLST,n)
DMAMSOUT EQU   3                          PATHNAM = *SYSOUT
*
DMAMSYSNUM DS    X                        Syslst number
DMAMRES  DS    XL3                       Alignment
DMAMENAM DS    CL64                      Elementname oder Blank
DMAMEVER DS    CL24                      Elementversion oder Blank
DMAMETYP DS    CL8                       Elementtyp oder Blank
DMAMRES2 DS    XL4                       Alignment
DMAM#    EQU   *-DMAMHDR
```

Returncode

Der Returncode wird im Standardheader des Parameterbereichs abgelegt. Der Parameterbereich darf dann nicht im Read-only-Bereich liegen, sonst erfolgt Programmterminierung.

Folgende Returncodes werden von Mail-File erzeugt:

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'00'	X'0000'	kein Fehler
X'00'	X'01'	X'0554'	Format des Dateinamens unzulässig
X'00'	X'01'	X'0576'	fehlerhafte Operandenkombination oder nicht gelöschte UNUSED-Felder
X'00'	X'20'	X'05C7'	interner Fehler im DMS
X'00'	X'40'	X'05FC'	Benutzerkennung nicht eingetragen
X'00'	X'40'	X'0694'	Senden der Datei nicht erlaubt
X'00'	X'40'	X'0695'	E-Mail-Adresse fehlt
X'00'	X'40'	X'0696'	E-Mail-Adresse von TSOS fehlt

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Außerdem können Returncodes von DMS-Schnittstellen (siehe Makro FSTAT bzw. ERASE) sowie von der Mail-Sender-Schnittstelle (Makro YMLSML des Software-Produkts interNet-Services) durchgereicht werden.

Wenn ein PLAM-Bibliothekselement (Operand LIBELEM) angegeben ist, dann können außerdem Returncodes von ILAM-Schnittstellen (siehe Makros PMATCH, PMDTCH, PMOPEN, PMCLOS, PMGETA, PMPOSA, PMDELM der ILAM-Schnittstelle für PLAM) durchgereicht werden. Die Maincodes der ILAM-Schnittstellen entsprechen den dezimalen Meldungsnummern der PLAM-Meldungen (also zum Beispiel Maincode 00CB entspricht der PLAM-Meldung PLA0203).
Nähere Informationen können Sie mit `/HELP-MSG <msgid>` abfragen.

Beispiel für eine Aufruffolge

```
         MVC   MLFLMFC(256),MLFLMFL
         MVC   MLFLMFC+256(XMAM#-256),MLFLMFL+256
         MAILFIL MF=M,PREFIX=X,DELETE=*DESTROY
         MAILFIL MF=E,PARAM=MLFLMFC
             .
             .
MLFLMFC  MAILFIL MF=C,PREFIX=X
MLFLMFL  MAILFIL MF=L,PATHNAM='PNO',USERID='UIO'
```

Selektion von E-Mail-Adressen über den Jobnamen

MAIL-FILE ermittelt die E-Mail-Adressen von Empfänger und Absender über den Benutzereintrag der jeweiligen Benutzerkennung. Zur Ausführung des Kommandos müssen die Benutzerkennungen des Empfängers und des Absenders jeweils eine E-Mail-Adresse enthalten (siehe Kommando SHOW-USER-ATTRIBUTES, Ausgabefeld EMAIL-ADDRESS). Der Eintrag kann auch eine Adressliste, d.h. mehrere durch Komma getrennte E-Mail-Adressen enthalten.

Bei einer Adressliste im Benutzereintrag des Absenders wird die erste Adresse zur Absenderadresse.

Bei einer Adressliste im Benutzereintrag des Empfängers unterscheidet MAIL-FILE, ob als Empfänger die Benutzerkennung des Aufrufers (*OWN) oder eine „fremde“ Benutzerkennung angegeben wurde. Bei Angabe einer fremden Benutzerkennung verschickt MAIL-FILE die E-Mail an alle Adressen. Bei Angabe der eigenen Benutzerkennung, selektiert MAIL-FILE die Adressen über den Jobnamen der aufrufenden Task:
Es wird eine Adresse gesucht, bei der ein Teilname des lokalen Adressteils (vor dem @) mit dem Jobnamen beginnt (Groß-/Kleinschreibung bleiben unberücksichtigt). Teilnamen sind durch einen Punkt von einander getrennt (z.B. vorname.nachname).

Aus der Adressliste `Anna.Huber@xy,Anja.Bauer@xy,Anton.Baumann@xy` werden z.B. folgende Adressen selektiert:

- `Anna.Huber@xy` mit den Jobnamen: ANN, HU, HUBER
- `Anja.Bauer@xy` mit den Jobnamen: ANJ, ANJA, BAUE, BAUER
- `Anton.Baumann@xy` mit den Jobnamen: ANT, BAUM, BAUMAN

Zusätzlich kann auch die Möglichkeit genutzt werden, dass den Adressen im Benutzereintrag „Adressnamen“ in runden Klammern vorangestellt werden.
Beispiel: `(ANH)Anna.Huber@xy,(ANB)Anja.Bauer@xy,(BMN)Anton.Baumann@xy`

Aus dieser Adressliste werden dann z.B. folgende Adressen selektiert:

– `Anna.Huber@xy` mit den Jobnamen: ANH sowie ANN, HU, HUBER
– `Anja.Bauer@xy` mit den Jobnamen: ANB sowie ANJ, ANJA, BAUE, BAUER
– `Anton.Baumann@xy` mit den Jobnamen: BMN sowie ANT, BAUM, BAUMAN

Wenn der Jobname zu mehr als einer Adresse passt, wird diejenige Adresse selektiert, bei der der zum Jobnamen passende Teilname am kürzesten ist. Aus der Adressliste `Beate.Pauli@xy,Pauline.Beck@xy,Paul.Becker@xy` werden z.B. folgende Adressen selektiert:

– `Beate.Pauli@xy` mit den Jobnamen: PAULI, BEA
– `Pauline.Beck@xy` mit den Jobnamen: PAULIN, BE, BECK
– `Paul.Becker@xy` mit den Jobnamen: P, PAUL, BECKER

Wenn bei mehreren Adressen der zum Jobnamen passende Teilname am kürzesten ist, wird von diesen Adressen die erste selektiert.

Wenn innerhalb einer Adresse mehrere Teilnamen zum Jobnamen passen, wird nur der erste Teilname berücksichtigt.

Wenn die aufrufende Task keinen Jobnamen besitzt oder der Jobname zu keiner Adresse der Adressliste passt, wird wie folgt vorgegangen:
– Bei Ermittlung der Empfängeradresse wird die ganze Adressliste verwendet, also die E-Mail an alle Adressen verschickt.
– Bei Ermittlung der Absenderadresse wird nur die erste Adresse der Adressliste verwendet.

NDWERINF – Status-Bytes abfragen

Im FCB werden im Fehlerfall die Status-Bytes (Sense-Bytes usw.) versorgt. Mit dem Makroaufruf NDWERINF kann der Anwender diese Status-Bytes auswerten, da NDWERINF Equate-Anweisungen für die logische Fehlerinformation generiert.

Hinweis

Der Makro NDWERINF sollte in Neuanwendungen nicht mehr verwendet werden, da er nur noch aus Kompatibilitätsgründen unterstützt wird. Die von ihm gelieferte Fehlerinformation ist im logischen Returncode des FCB enthalten und kann dort ausgewertet werden.

Format

Operation	Operanden
NDWERINF	[{ prefix / * }][,ONLYOSB=YES]

Operandenbeschreibung

prefix
Präfix für die generierten Namen (1 Buchstabe).

Voreinstellung: I

*****
Es wird kein Präfix generiert.

ONLYOSB=YES
Es werden nur Equates für die drei Sense-Bytes (OSB) generiert.

OPEN – Datei eröffnen

Makrotyp: R-Typ

Mit dem Makroaufruf OPEN muss jede Datei vor der Bearbeitung eröffnet werden. Die Voreinstellung für den OPEN-Makroaufruf ist durch die Angabe im OPEN-Operanden von FCB oder FILE (über die TFT) gegeben; ist dort keine Angabe zum OPEN-Modus gemacht, gilt OPEN=INPUT.

Format

Operation	Operanden
OPEN	{ fcbadr | (1) }, { mode | (0) }[,PARMOD={ 24 | 31 }]

Operandenbeschreibung

fcbadr
Adresse des FCB-Makroaufrufs für die Datei, die eröffnet werden soll.

(1)
Die FCB-Adresse steht im Register 1.

mode
OPEN-Modus, siehe Tabelle unten: Spalte 1

(0)
Der OPEN-Modus wird im Register 0 übergeben, codiert im niedrigstwertigen Byte.
Die folgende Tabelle zeigt die Zuordnung von OPEN-Modi und Codes.

OPEN-Modus	Code	Bedeutung
unbestimmt	X'00'	der im TU-FCB angegebene OPEN-Modus hat Vorrang. Ist auch dieser X'00', wird eine Fehlermeldung ausgegeben.
INPUT	X'01'	die Datei wird als Eingabedatei eröffnet, anschließend sind nur Leseoperationen zulässig; Leserichtung: „vorwärts“ Dateianfang → Dateiende
REVERSE	X'02'	die Datei wird als Eingabedatei eröffnet, anschließend sind nur Leseoperationen zulässig; Leserichtung: „rückwärts“ Dateiende → Dateianfang

OPEN-Modus	Code	Bedeutung
OUTPUT	X'04'	die Datei wird als Ausgabedatei eröffnet und, falls sie schon existierte, von Beginn an überschrieben; es ist nur sequenzielles Schreiben zulässig
EXTEND	X'08'	die Datei wird als Ausgabedatei eröffnet zur sequenziellen Erweiterung; es sind, beginnend an einem definierten Punkt der Datei, nur sequenzielle Schreiboperationen zulässig Bei einer leeren Datei wird OPEN EXTEND auf OPEN OUTPUT abgebildet.
UPDATE	X'10'	die Datei soll aktualisiert werden; es sind sowohl Lese- als auch Schreiboperationen zulässig
INOUT	X'20'	die Datei soll aktualisiert werden; es sind sowohl Lese- als auch Schreiboperationen zulässig
OUTIN	X'40'	die Datei soll zunächst erstellt (bzw. überschrieben) werden; anschließend sind sowohl Lese- als auch Schreib-Operationen zulässig

PARMOD
gibt den Generierungsmodus an, entsprechend dem im FCB-Makroaufruf für die Datei gültigen Wert.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert.

= 24
Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (nur 24-Bit-Adressierung).

= 31
Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist (24-Bit- oder 31-Bit-Adressierung).

Hinweise zur Programmierung

1. Der OPEN-Makroaufruf zerstört die Register 0, 1, 14 und 15.
2. Wird während eines Programmlaufs bei einem OPEN-Aufruf für eine Datei eine FCB-Adresse angegeben, die bereits für einen anderen erfolgten OPEN verwendet wurde, wird das Programm abnormal mit Fehlerschlüssel DMS0D9F beendet.

Hinweise zur Programmierung bzgl. großer Dateien

Der Makro OPEN und die Zugriffsmethoden sind nur von der Einführung großer Dateien, nicht aber von der Einführung großer Volumes betroffen.

Die Schnittstelle OPEN prüft, ob Dateierweiterungen über 32 GB hinaus und das Erstellen oder Zugriffe auf Dateien ≥ 32 GB zulässig sind.

Hierbei gibt es zwei Aspekte:

a) Abweisen des Zugriffs auf oder der Erzeugung von großen Dateien für Zugriffsmethoden, die eine Bearbeitung von großen Dateien nicht gestatten.

b) Kennzeichnen, dass ein Programm Dateien ≥ 32 GB erzeugen bzw. öffnen kann.

Unverträgliche Schnittstellenvarianten

Schnittstellen, an denen 3-Byte-Blocknummern verwendet werden, sind prinzipiell nicht in der Lage, mit Dateien ≥ 32 GB zu arbeiten. Es handelt sich hier um folgende Fälle:

– Sämtliche Dateien im Key-Format (BLKCTRL=PAMKEY):
  Die logischen Blocknummern im Pamkey sind nur 3 Byte breit.
– 24-Bit-Schnittstelle von UPAM:
  Das Feld für die logischen Blocknummern in den UPAM-Parameterlisten und im TU-FCB ist nur 3 Byte breit.
– 24-Bit-Schnittstelle von SAM:
  Hier sind die logischen Blocknummern als Teil der Wiedergewinnungsaddresse betroffen.

In allen oben aufgeführten Fällen gilt:

– Der Zugriff auf Dateien ≥ 32 GB wird mit dem Returncode `X'0000D9D'` oder `X'00000D00'` abgewiesen, abhängig von der Größe des für die Datei allokierten Speicherplatzes (FILE-SIZE).
– Die Überschreitung einer Dateigröße von 32 GB durch Sekundärallokierung wird unterbunden (LARGE-FILE).

Semantische Inkompatibilität

Es kann nicht ausgeschlossen werden, dass Anwendungen zwar Schnittstellen benutzen, die bezüglich der oben angeführten Datenfelder bereits 4-Byte-Felder verwenden, ihrerseits jedoch explizit oder implizit von der bisherigen Beschränkung auf Werte kleiner X'00FFFFFF' Gebrauch machen.
Beachten Sie deshalb, dass bei der Anpassung von Assemblercode für Dateien ≥ 32 GB neben der Umstellung auf 4-Byte-Blockzähler und -nummern auch zu prüfen ist, ob die Programmlogik implizit von der Annahme Gebrauch macht, dass Dateien nicht größer als 32 GB werden können.

Ohne Anspruch auf Vollständigkeit folgen einige Beispiele für derartige semantische Inkompatibilitäten:

– Die höchste 3-Byte-Blocknummer X'FFFFFF' hat eine spezielle Bedeutung.

– „Blocknummern“ > X'00FFFFFF' repräsentieren nicht Blöcke, sondern andere Objekte.

– Bei Berechnungen mit Blocknummern oder - zählern > X'00FFFFFF' kann es zum Überlauf kommen.

– Die Stellenzahl von Ein- oder Ausgabefeldern reicht nicht zur Darstellung beliebig großer Blocknummern oder -zähler.

– Bei Umrechnungen von Hexadezimalzahlen in Dezimalzahlen ist die Feldlänge für die Dezimalzahl zu klein.

– Es wird unterstellt, dass Datenstrukturen, deren Umfang von einer Dateigröße abhängt, stets im virtuellen Speicher Platz finden. Diese Annahme kann für Dateien < 32 GB gültig sein, nicht aber, wenn diese Größe überschritten wird.

Eine abschließende Liste von Problemfällen kann nicht gegeben werden. Potenziell kritisch ist jedenfalls Coding, das auf Blocknummern, Dateigrößen und davon abgeleiteten oder abhängigen Strukturen aufsetzt.

Über die Ausführung des Makros bzgl. großer Dateien informieren folgende Returncodes:

X'cc'	**X'bb'**	**X'aaaa'**	**Erläuterung**
X'00'	X'00'	X'0D9D'	Fehler beim Eröffnen einer Plattendatei.
X'00'	X'00'	X'0D00'	Systemfehler beim Eröffnen einer Datei.

OSTAT – Information über eröffnete Dateien anfordern

Makrotyp: R bei PARMOD=24
0 bei PARMOD=31

Mit dem OSTAT-Makroaufruf kann sich der Anwender darüber informieren

– wie viele Aufträge die Datei eröffnet haben
– wie viele Aufträge die Datei für ISAM-Shared-Update eröffnet haben (in allen möglichen OPEN/SHARUPD-Kombinationen)
– wie viele Aufträge die Datei mit einer anderen Zugriffsmethode als ISAM eröffnet haben

Für den Ausgabebereich (Operand area) kann mit dem Makro IDOST eine DSECT erzeugt werden.

Die Datei, für die der OSTAT-Makro aufgerufen wird, muss vorher eröffnet sein. Die OPEN/SHARUPD-Kombination des aufrufenden Auftrags ist dann in der Ausgabe enthalten!

Format

Operation	Operanden
OSTAT	$\left\{\begin{array}{l}\texttt{fcbadr}\\\texttt{(1)}\end{array}\right\},\left\{\begin{array}{l}\texttt{area}\\\texttt{(0)}\end{array}\right\}[,\texttt{PARMOD=}\left\{\begin{array}{l}\texttt{24}\\\texttt{31}\end{array}\right\}]$

Operandenbeschreibung

fcbadr
Adresse des FCB der Datei, für die der OPEN-Status ausgegeben werden soll

(1)
Die FCB-Adresse steht im Register 1.

area
Bereich, in den die Information ausgegeben werden soll. Dieser Bereich besteht aus 23 Zählerfeldern, die jeweils 1 Byte lang sind. Der maximale Wert für einen einzelnen Zähler ist daher 255. Nimmt einer der ISAM-internen Zähler einen größeren Wert an, wird der entsprechende Zähler im Informationsbereich auf 255 gesetzt.

Mit dem Makro IDOST kann eine DSECT für diesen Bereich erzeugt werden.

Wird der Operand „area“ nicht angegeben, wird der Bereich für die Information im Makro generiert, sodass ein nicht reentrant-fähiger Code entsteht.

(0)
Die Adresse des Bereiches, in dem die Information abgesetzt werden soll, steht im Register 0.

PARMOD
Gibt den Generierungsmodus an.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24
Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (nur 24-Bit-Adressierung).

= 31
Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist (24-Bit- oder 31-Bit-Adressierung).

Hinweis zur Programmierung

Der OSTAT-Makroaufruf zerstört die Register 0, 1, 14 und 15.

Rückinformation

Register 1	Adresse des FCB
Register 0	Adresse des Ausgabebereichs „area“.
Register 15	in den beiden niederwertigen Bytes X'0000' (= Makroaufruf erfolgreich ausgeführt) oder Fehlerschlüssel.

PAM – UPAM-Aktionen ausführen

Makrotyp: S-Typ

Alle Benutzeranforderungen an das DVS bezüglich UPAM-Aktionen werden über diesen Makroaufruf abgewickelt.

Format

Operation	Operanden
PAM	fcbadr[,PARMOD={24 | 31}] [,{RDWT | CHK | LOCK | LRD | LRDWT | RD | RDEQU | SETL | SETLPP | SYNC | UNLOCK | WRT | WRTWT | WRTWU | WT}][,CHAIN=relaus][,FECB=relaus][,HP={zahl | +zahl | -zahl}] [,KEYFLD=relaus][,LEN={STD | (STD,n) | länge}][,LOC={1 | 2 | relaus}] [,MKEY={NO | YES}][,REQNO=zahl][,POST=postcode]

Operandenbeschreibung

fcbadr
Gibt die Adresse des FCB an, der mit dieser Datei in Verbindung steht.

CHAIN = relaus
Bezeichnet die symbolische Adresse des nächsten Elements einer Kette von PAM-Makroaufrufen in Listenform. Eine solche Kette kann höchstens 255 Elemente enthalten.

Voreinstellung: die Liste wird als das letzte Element einer Kette betrachtet

CHK
Prüft, ob die angegebene Ein-/Ausgabe-Anforderung abgeschlossen ist. Ist dies der Fall, läuft das Programm weiter. Andernfalls wird die Steuerung an die Adresse übergeben, die im Anwenderprogramm unter dem LOC-Operanden angegeben ist.

FECB = relaus
Symbolische Adresse eines Ereignissteuerblocks (siehe Seite 106). Der Operand darf nur bei den Operationen RD, LRD und WRT angegeben werden.

HP
verweist auf eine PAM-Seite, und zwar
– bei nichtgeketteter Ein-/Ausgabe auf die zu übertragende (oder zu sperrende) PAM-Seite;
– bei geketteter Ein-/Ausgabe (für Plattendateien oder für Banddateien mit Nichtstandardblöcken (NK-Dateien)) auf die Erste einer Reihe zu übertragender (oder zu sperrender) PAM-Seiten.

Voreinstellung: HP=+1 (d.h. sequenzielle Verarbeitung)

Hinweis

Bei PAM- und SAM-Dateien ohne PAM-Schlüssel (Eigenschaft BLKCTRL=DATA oder BLKCTRL=NO) gelten folgende Einschränkungen, die sich daraus ergeben, dass in diesen Dateien nur auf Logische Datenblöcke und nicht auf jeden 2K-Block zugegriffen werden kann:

– Bei den Operationen LRD, LRDWT, RD, RDWT, RDEQU, WRT, WRTWT, WRTWU, LOCK und UNLOCK muss HP auf den Beginn eines Logischen Blockes zeigen, d.h.
HP = n * BLKSIZE + 1 (n ≥ 0). Andernfalls wird der PAM-Aufruf abgewiesen.

– Bei den Operationen SETL und SETLPP muss HP auf das Ende eines Logischen Blockes zeigen, d.h. HP = n * BLKSIZE (n ≥ 0). Andernfalls wird der PAM-Aufruf abgewiesen. (Im Fall SETL ist der Dateizeiger damit für den Zugriff auf den **nächsten** Logischen Block vorbereitet.)

Näheres ist den „Hinweisen zur Programmierung“ am Ende der Beschreibung des PAM-Makroaufrufs auf Seite 765 zu entnehmen.

= zahl
logische Nummer der PAM-Seite (LHP) innerhalb der Datei.

= +zahl
= -zahl
Die logische Seitennummer bezieht sich auf den Dateizeiger. Die entsprechende absolute logische Seitennummer wird als Summe von Dateizeiger und angegebener relativer Seitenummer errechnet.

KEYFLD = relaus
Wird nur bei Dateien mit PAM-Schlüssel ausgewertet (Eigenschaft (BLKCTRL=PAMKEY) und bei Dateien mit BLKCTRL=DATA oder BLKCTRL=NO ignoriert.
Der Operand gibt bei nichtgeketteter Ein-/Ausgabe die Adresse eines 16 Byte langen Bereiches an, in den beim Lesen der PAM-Schlüssel gebracht bzw. aus dem beim Schreiben der PAM-Schlüssel gebildet wird.

Bei geketteter Ein-/Ausgabe mit separater Verarbeitung aller zugehörigen PAM-Schlüssel (Operand MKEY=YES) muss hier die Adresse eines genügend großen Bereichs angegeben werden (Anzahl zu übertragender PAM-Seiten mal 16 Byte). Soll nur der PAM-Schlüssel der ersten PAM-Seite verarbeitet werden (MKEY=NO), muss der angegebene Bereich nur 16 Byte lang sein. In diesem Fall kann der Operand KEYFLD entfallen, und es tritt der Standardwert in Kraft.

Der Operand KEYFLD bleibt unberücksichtigt bei den Operationen WT, CHK, SETL und SYNC.

Voreinstellung: (bei Dateien mit BLKCTRL=PAMKEY)
– Adresse des ID1KEY1-Feldes im FCB bei allen Lese- und Schreiboperationen außer RDWT, LRDWT und RDEQU
– Adresse des ID1KEY2-Feldes im FCB bei RDWT, LRDWT und RDEQU

Weiteres siehe „Hinweise zur Programmierung“ auf Seite 765.

LEN
Legt die Länge fest, in der bei einem PAM-Aufruf Daten übertragen werden

Voreinstellung: LEN=STD

Bei den Operationen WT, CHK und SETL wird der Operand LEN nicht berücksichtigt.

Für den Zusammenhang mit der Blocklänge (BLKSIZE) siehe UPAM-FCB-Makroaufruf (Seite 91).

Weiteres siehe „Hinweise zur Programmierung“ auf Seite 765.

= STD
Die Daten werden in der Länge eines Standardblockes (≙ 2048 Byte) übertragen.

= (STD,n)
Die Daten werden in der Länge von n Standardblöcken (zu je 2048 Byte) übertragen. n ist eine ganze Zahl mit $1 \leq n \leq 16$.
Der Operandenwert kann nur für die 31-Bit-Schnittstelle des Makros angegeben werden.

= länge
Länge der zu übertragenden Daten in Byte ($1 \leq$ länge ≤ 32768). Dabei sind folgende Fälle zu unterscheiden:
– $1 \leq länge \leq 2048$: keine gekettete Verarbeitung; je PAM-Makroaufruf wird ein Block der Länge 2048 Byte vom/zum Puffer übertragen, d.h. geschrieben oder gelesen.
– $2049 \leq länge \leq 32768$: gekettete Verarbeitung.

Bei Platten- und Banddateien mit PAM-Schlüssel wird die Anzahl der mit einem PAM-Aufruf zu übertragenden PAM-Blöcke folgendermaßen ermittelt:
– Ist *länge* ein ganzzahliges Vielfaches von 2048 (länge = n*2048, $n \leq 16$), gibt der Quotient n = länge/2048 die Anzahl der zu einer Übertragungseinheit gehörenden PAM-Blöcke an.
– Ist *länge* kein ganzzahliges Vielfaches von 2048, wird der Quotient auf die nächsthöhere ganze Zahl gerundet. Bei bestimmter Hardware kann eine solche Angabe zur Zwischenpufferung und damit zu Performance-Einbußen führen.

Grundsätzlich wird im Rahmen des CLOSE bei Dateien mit BLKCTRL=PAMKEY die Position des letzten gültigen Bytes im Last Byte Pointer des letzten PAM-Blocks und bei Dateien mit BLKCTRL=NO oder DATA im Last Byte Pointer des letzten logischen Blocks der Datei gespeichert.

Bei Node-Files werden am Dateiende nur die Daten in der Länge *länge* übertragen. Damit endet ein Node-File bei anschließendem CLOSE auf Byte-Grenze.

Beim Lesen werden die Daten nur in der Länge *länge* übertragen, beim Schreiben sind sie nur in dieser Länge gültig.

Bei PAM- oder SAM-Plattendateien **ohne PAM-Schlüssel** wird bei der Ermittlung des Dateizeigers die Länge eines Logischen Datenblockes berücksichtigt; es werden jedoch immer nur so viele Blöcke der Länge 2048 Byte geschrieben, wie es die Angabe im Operanden LEN erfordert.

LOC
Verweist auf den Ein-/Ausgabepuffer im Arbeitsspeicher (gilt nicht für CHK, siehe unter „=relaus“). Der Puffer muss mindestens so viele PAM-Seiten aufnehmen können, wie durch den Operanden LEN= festgelegt ist. Die Pufferadresse kann beliebig auf Bytegrenze ausgerichtet sein. Ist der Puffer ≤ 4096 Byte (Hauptspeicher-)Seite), sollte er in einer Seite enthalten und auf Wortgrenze ausgerichtet sein. Ist dieser Bereich > 4096 Byte, sollte er auf Seitengrenze ausgerichtet sein. Wenn diese Ausrichtung nicht beachtet wird, kann dies bei bestimmter Hardware Zwischenpufferung und damit Performance-Einbuße zur Folge haben.

Voreinstellung:
- IOAREA2-Adresse im FCB, falls zuvor schon ein PAM-Aufruf für diesen FCB ausgeführt wurde und beim letzten PAM-Aufruf IOAREA1 angegeben wurde.
- IOAREA1-Adresse im FCB in allen anderen Fällen

Der Operand LOC bleibt unberücksichtigt bei den Operationen WT, LOCK, UNLOCK, SETL, SETLPP und SYNC.

= 1
Die Pufferadresse steht im FCB-Feld IOAREA1.

Diese Angabe ist nicht erlaubt, wenn im OPEN-Makro IOAREA1=NO angegeben wurde. Von der (voreingestellten) Möglichkeit des Pufferwechsels darf nur Gebrauch gemacht werden, wenn im OPEN weder IOAREA1=NO noch IOAREA2=NO angegeben wurde.

= 2
Die Pufferadresse steht im FCB-Feld IOAREA2.

Diese Angabe ist nicht erlaubt, wenn im OPEN-Makro IOAREA2=NO angegeben wurde. Von der (voreingestellten) Möglichkeit des Pufferwechsels darf nur Gebrauch gemacht werden, wenn im OPEN weder IOAREA1=NO noch IOAREA2=NO angegeben wurde.

= relaus
Gibt die Pufferadresse bzw. bei CHK die Fortsetzungsadresse an, falls eine geprüfte Ein-/Ausgabe-Anforderung nicht abgeschlossen ist.

Bei der Operation CHK muss „LOC=relaus“ angegeben werden; d.h. eine Adresse, an der der Prozess fortgesetzt werden soll, wenn die geprüfte Ein-/Ausgabeoperation noch nicht abgeschlossen ist.

LOCK
Nur für Plattendateien:
fordert Sperren an für einen oder mehrere PAM-Blöcke (siehe auch Operanden HP, LEN). Sperrung eines Blockes bedeutet, dass andere PAM-Aufrufe mit LOCK oder LRD bzw. LRDWT für diesen Block abgewiesen werden.

LRD
Nur für Plattendateien:
wie LOCK; wurde eine Sperrung erreicht, ist der weitere Verlauf wie bei RD.

LRDWT
Nur für Plattendateien:
wie LOCK und LRD; wurde eine Sperrung erreicht, ist der weitere Verlauf wie bei RDWT.

MKEY
Nur für Plattendateien:
dieser Operand hat nur bei Dateien mit PAM-Schlüssel (Eigenschaft BLKCTRL=PAMKEY) und bei geketteter Ein-/Ausgabe Bedeutung (vgl. auch Operand KEYFLD).

Voreinstellung: MKEY=NO

= NO
Es wird nur der PAM-Schlüssel der Ersten einer Reihe von PAM-Seiten vom Benutzer erwartet bzw. bereitgestellt.

= YES
PAM-Schlüssel werden vom Benutzer für jede gelesene PAM-Seite erwartet bzw. für jede zu schreibende PAM-Seite bereitgestellt.

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.

Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst.
Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

POST = postcode
Wird in Zusammenhang mit PARMOD=31 TU-Eventing genutzt, kann der Anwender hier einen POST-Code definieren, ansonsten wird die Angabe eines POST-Wertes ignoriert.

Der POST-Code ist je nach SOLSIG-Definiton bei Beendigung der Ein-/Ausgabe wiederzufinden:
– in den rechten beiden Byte des Feldes RPOSTAD
– in den rechten beiden Byte des bei RPOSTAD angegebenen Registers
– in den rechten beiden Byte des Registers 3 des Contingency-Prozesses, der durch das UPAM-Event gestartet wurde (Näheres siehe Handbuch „Makroaufrufe an den Ablaufteil“ [2], Makro SOLSIG).

RD
Stößt das Lesen eines Datenblocks von der Datei in den Hauptspeicher an; der Auftrag wird unmittelbar nach dem Anstoßen der Leseoperation fortgesetzt.

RDEQU
wie RDWT; bei Plattendateien mit **D**ual **R**ecording by **V**olume (DRV, siehe Handbuch „DRV“ [15]) wird zusätzlich die Kopie aktualisiert.
Der Operandenwert kann nur für die 31-Bit-Schnittstelle des Makros angegeben werden.

RDWT
wie RD; der Auftrag läuft jedoch erst nach Beendigung der Leseoperation weiter.

REQNO = zahl
Nur für Plattendateien:
gibt die Nummer der Ein-/Ausgabe-Anforderung an, die dieser Operation zugeordnet ist. Wird dieser Operand angegeben, darf der Operand FECB nicht angegeben werden.

Voreinstellung: REQNO=1

SETL
Bewirkt, dass der Dateizeiger auf die angegebene PAM-Seite eingestellt wird.

SETLPP
Nur für Plattendateien:
bewirkt, dass der Last Page Pointer (Dateiende-Zeiger) auf die angegebene PAM-Seite gesetzt wird; sie muss bereits zu der Datei gehören. Diese Operation ist nicht zulässig für Eingabedateien (OPEN INPUT) und Dateien, die mit SHARUPD=WEAK oder YES eröffnet wurden.
Node-Files werden nach SETLPP abgeschnitten („truncate“).

SYNC
wartet auf Ein-/Ausgabe-Beendigung und leert den Steuerungspuffer bei Magnetbandkassetten; für Dateien, die nicht auf Magnetbandkassetten stehen, entspricht SYNC dem WAIT. Der Operandenwert kann nur für die 31-Bit-Schnittstelle des Makros angegeben werden.

UNLOCK
Nur für Plattendateien:
gibt gesperrte PAM-Blöcke frei (siehe auch Operanden HP und LEN).

WRT
Stößt das Schreiben eines Datenblocks aus dem Hauptspeicher in die Datei an; unmittelbare Fortsetzung des Auftrags.

WRTWT
wie WRT; Fortsetzung des Auftrags jedoch erst nach Beendigung der Schreiboperation.

WRTWU
wie WRTWT; unmittelbar nach dem Abschluss der Ein-/Ausgabe-Operation wird die soeben geschriebene PAM-Seite freigegeben.

WT
Bewirkt, dass das Programm auf den Abschluss einer bestimmten Ein-/Ausgabe-Anforderung wartet. Nach erfolgtem Abschluss läuft das Programm weiter.

Hinweise zur Programmierung

1. Der PAM-Makroaufruf zerstört die Register 0, 1, 14 und 15.
2. Die erste PAM-Seite einer PAM-Datei hat die Nummer 1. Bei der Eröffnung einer PAM-Datei erhält der Dateizeiger jedoch den Anfangswert 0. Der HP-Operand bleibt bei den Operationen WT bzw. CHK und SYNC unbeachtet. WT bzw. CHK beziehen sich nicht auf PAM-Seiten, sondern auf Ein-/Ausgabe-Anforderungen.
3. Jede Operation (ausgenommen WT, CHK, SYNC sowie SETLPP auf Banddatei), bei der keine Fehlerroutine angesprungen wird, bewirkt, dass der Dateizeiger die Nummer der PAM-Seite erhält, auf die soeben zugegriffen wird. Dies gilt auch für Sperrungen und Freigaben, auch bei SHARUPD=NO. Der Dateizeiger bleibt grundsätzlich unverändert, wenn Operationen nicht normal abgeschlossen werden, sondern zu einer Fehlerroutine führen.

   Bei PAM- und SAM-Dateien ohne PAM-Schlüssel (Eigenschaft BLKCTRL=DATA oder BLKCTRL=NO) enthält der Dateizeiger anschließend die Nummer des letzten 2-KB-Blockes in dem Logischen Datenblock, auf den zuletzt zugegriffen wurde.
   (Bei der Berechnung des Dateizeigers wird also die Länge auf das nächstgrößere Vielfache von BLKSIZE aufgerundet.)
4. Bezeichnet für eine Plattendatei n die Anzahl derjenigen PAM-Seiten, die einer Datei bereits zugewiesen sind, und k den Wert der Sekundärzuweisung, so sind folgende Regeln zu beachten:

PAM-Operation	**Bedeutung**
RD RDEQU RDWT LRD LRDWT	HP = zahl, 1 ≤ zahl ≤ n
WRT WRTWT WRTWU	HP = zahl, 1 ≤ zahl ≤ n + k sobald eine PAM-Seite mit LHP > n geschrieben wird, erfolgt eine Sekundärzuweisung; der Umfang der Datei wächst auf n + k PAM-Seiten sollen mehrere aufeinander folgende PAM-Seiten mit einem PAM-Makroaufruf geschrieben werden (gekettete Ein-/Ausgabe), muss (spätestens) nach der ersten Sekundärzuweisung Speicherplatz für alle zu schreibenden Blöcke zugewiesen sein, d.h. für den letzten zu schreibenden Block gilt: LHP ≤ n+k
LOCK UNLOCK	HP = zahl, zahl > 0 es können später PAM-Seiten gesperrt oder freigegeben werden, die momentan nicht zugewiesen sind; eine Sperre/ Freigabe ist nicht mit einer Sekundärzuweisung verbunden

5. Der mit REQNO genannte Wert muss kleiner oder gleich dem im PAMREQS-Operanden des FCB genannten Wert sein.

6. Der Operand REQNO bleibt unberücksichtigt bei den Operationen LOCK, UNLOCK und SETL. Bei den Operationen WT, CHK und SYNC wird damit angegeben, für welche Ein-/Ausgabe-Anforderung die Operation ausgeführt werden soll.
7. Jeder einzelnen Anforderungsnummer (REQNO) kann nur eine asynchrone Ein-/Ausgabe-Operation zugeordnet werden. Die Beendigung dieser asynchronen Ein-/Ausgabe-Operation kann entweder explizit durch Angabe ihrer Anforderungsnummer in einer WT-Operation oder implizit durch eine neue Lese- oder Schreiboperation an die gleiche Anforderungsnummer abgewartet werden. Wird beim impliziten Warten auf die Beendigung einer Ein-/Ausgabe-Anforderung ein Fehler entdeckt, so lautet der Fehlercode '997', und die Anforderung, die den impliziten WT verursachte, wird nicht ausgeführt.
8. Anforderungsnummern (REQNO) sind Betriebsmittel des Systems; ein Missbrauch kann die Leistungsfähigkeit des Systems mindern.
9. Will der Benutzer für Ein- / Ausgabeoperationen den vom OPEN angelegten Datenpuffer verwenden (Operand LOC=1 oder LOC=2), so darf der Wert des Operanden LEN nicht größer sein als BLKSIZE, da sonst der Puffer nicht zur Aufnahme der Daten ausreichen würde.
10. Bei Banddateien muss der Wert für LEN genau die Länge eines Bandpuffers angeben. Bei Dateien mit BLKCTRL=PAMKEY ist dies ein PAM-Block (2048 Byte); dies entspricht der Angabe LEN=STD bzw. LEN=(STD,1) oder LEN=2048 (16 Byte für den PAM-Schlüssel werden vom System hinzugefügt).
11. Bei Banddateien ohne PAM-Schlüssel (Eigenschaft BLKCTRL=DATA oder BLKCTRL=NO) wird ein Nichtstandardblock in der mit LEN vereinbarten Länge geschrieben; beim Lesen wird versucht, einen Block in der mit LEN vereinbarten Länge zu lesen. Bei PAM- oder SAM-Dateien ohne PAM-Schlüssel darf der Wert für LEN nicht größer sein als BLKSIZE, bei ISAM-Dateien ohne PAM-Schlüssel nicht größer als 2048 bzw. (STD,1).
12. Ist beim Lesen von Banddateien ohne PAM-Schlüssel der Wert für LEN kleiner als die Blocklänge der Datei, so wird die Leseoperation mit Fehler beendet. Um solche Fehler zu vermeiden, sollte man eine Banddatei ohne PAM-Schlüssel stets mit LEN=BLKSIZE lesen, also mit dem größten zulässigen Wert für LEN. Bei Dateien mit BLKCTRL=DATA kann dann die Länge der gültigen Daten aus dem Blockkontrollfeld ermittelt werden.
13. Bei einer Schreiboperation (WRT, WRTWT, WRTWU) auf eine Datei mit BLKCTRL=DATA behandelt UPAM die ersten 12 Byte als Blockkontrollfeld und überschreibt sie mit Blockkontrollinformation.

14. Bei Dateien mit BLKCTRL=DATA und Operationen, die eine Ein-/Ausgabe anfordern, muss der Wert für LEN so gewählt werden, dass das Blockkontrollfeld stets vollständig im Datenpuffer liegt. Dies ist der Fall, wenn folgende Bedingungen erfüllt sind:
    - Bei PAM- und SAM-Plattendateien:
      LEN = n*BLKSIZE oder LEN > n*BLKLSIZE + 12 (n: Anzahl der Datenblöcke)
    - Bei ISAM-Plattendateien:
      LEN = n*2048 oder LEN > n*2048 + 12 (n: Anzahl der Datenblöcke)
    - Bei Banddateien: LEN > 12

15. Für Dateien mit BLKCTRL=DATA gilt folgende *Parallelitätseinschränkung*:
    Bei der Verarbeitung einer PAM-Datei mit BLKCTRL=DATA dürfen die IOAREAs nicht parallel für mehrere I/O-Aufträge verwendet werden, weil dabei der Inhalt des Blockkontrollfeldes undefiniert wäre. Beispiel:
    - Unzulässig ist:
      ```
      PAM   WRT,FCB1,LOC=BUFFER
      PAM   WRT,FCB2,LOC=BUFFER       (Parallele I/Os auf BUFFER)
      PAM   WT,FCB1
      PAM   WT,FCB2
      ```
    - Zulässig ist:
      ```
      PAM   WRTWT,FCB1,LOC=BUFFER     (Nacheinander ausgeführte
      PAM   WRTWT,FCB2,LOC=BUFFER     I/Os auf BUFFER)
      ```

16. Bei LOCK- oder UNLOCK-Operationen wird LEN=0 behandelt wie LEN=2048 (oder LEN=n mit $1 \leq n \leq 2048$). Bei jedem anderen Wert für LEN werden so viele PAM-Blöcke gesperrt, wie bei bei einer Ein-/Ausgabeoperation mit dieser LEN-Angabe gelesen oder geschrieben würden.
    Bei PAM- und SAM-Dateien ohne PAM-Schlüssel (Eigenschaft BLKCTRL=DATA oder BLKCTRL=NO) wird intern jeweils nur der erste 2-KB-Abschnitt jedes betroffenen logischen Datenblockes gesperrt, was jedoch die Sperrung des gesamten Datenblockes bewirkt.

17. Bei Dateien mit PAM-Schlüssel (BLKCTRL=PAMKEY) ist zu beachten:
    - Wird ein WT für eine erfolgreiche Leseoperation gegeben (explizit oder implizit), so wird der 16 Byte lange PAM-Schlüssel, der dem gelesenen Block zugeordnet ist, in das mit KEYFLD bezeichnete Feld gebracht. Dabei bewirkt eine CHK-Operation, die nach einer abgeschlossenen Ein-/Ausgabe-Operation gegeben wird, das Gleiche wie eine WT-Operation.
    - Bringt ein Anwender bei einer Schreiboperation irgendeine Information in die letzten 8 Byte des von diesem Operanden angegebenen 16-Byte-Feldes, so wird diese Information in die Datei geschrieben als Teil des PAM-Schlüssels, der der zu schreibenden PAM-Seite zugeordnet ist (nicht empfohlen). UPAM erstellt grundsätzlich die ersten 8 Byte dieses Feldes vor Beginn des Schreibvorgangs.
    - Der Operand KEYFLD bleibt unberücksichtigt bei den Operationen WT, CHK, UNLOCK, SETL und SETLPP.

PUT – Satz schreiben

ISAM:	Makrotyp:	R bei PARMOD=24 O bei PARMOD=31
SAM:	Makrotyp:	R bei PARMOD=24 und bei PARMOD=31

Anwendungsbereich

Die Funktion PUT ist bei der Dateiverarbeitung mit SAM oder ISAM (satzorientierte Zugriffsmethoden) möglich. Die Datei muss dabei mit einem der folgenden OPEN-Modi eröffnet worden sein:

EXTEND	bei SAM und ISAM
INOUT/OUTIN	bei ISAM
OUTPUT	bei SAM

Funktion

Der PUT-Makroaufruf dient zum sequenziellen Erstellen oder Erweitern einer Datei. Er verlängert die Datei um einen Satz.

Nur für K-ISAM:

Der PUT-Makroaufruf kann nur verwendet werden, wenn sequenziell an das Dateiende geschrieben werden soll. Wurde eine Datei INOUT oder OUTIN eröffnet, wird mit PUT der Satz an das aktuelle Dateiende geschrieben. War der letzte Makroaufruf zu dieser Datei kein PUT, wird mit dem PUT ein SETL an das Ende der Datei durchgeführt, bevor der Satz in die Datei geschrieben wird.

ISAM:

Beim sequenziellen Erstellen bzw. Erweitern einer Datei wird der PAD-Faktor (Blockfüllung) berücksichtigt. Die Auswirkungen bei NK-ISAM und K-ISAM sind jedoch unterschiedlich (Näheres siehe Seite 76).

Der PUT-Makroaufruf stellt sicher, dass der Datei nur Sätze hinzugefügt werden, deren Schlüssel gleich oder größer ist als der vorhandene größte Schlüssel. Wird die aufsteigende Folge der Schlüssel unterbrochen, geht die Steuerung an einen der beiden folgenden EXLST-Ausgänge:

– DUPEKY, falls der Schlüssel des aktuellen Satzes gleich dem Schlüssel des Vorgängers ist und nicht DUPEKY=YES angegeben ist
– SEQCHK, falls der Schlüssel des aktuellen Satzes kleiner ist als der höchste Schlüssel der vorhandenen Datei

NK-ISAM:

Ist für NK-ISAM im FCB IOAREA1=NO definiert, führt jeder PUT zu einem SVC, der PAD-Wert wird nicht berücksichtigt.

Ist IOAREA1≠NO, wird der Ein-/Ausgabebereich IOAREA1 so lange mit Sätzen gefüllt, bis eine der folgenden Bedingungen erfüllt ist:

– der Puffer ist gefüllt
– die mit PAD in FILE/FCB definierte Grenze ist überschritten
– ein anderer ISAM-Aktionsmakro als PUT wird ausgeführt

Dann wird ein SVC abgesetzt und der Datei ein neuer Datenblock mit dem Inhalt des Ein-/Ausgabebereichs hinzugefügt.

Bei Shared-Update-Verarbeitung ist nach einem PUT der gesamte Bereich zwischen dem aktuellen höchsten Schlüssel und dem Dateiende gesperrt.

Locate-Mode (Ortungsbetrieb):

Im Locate-Mode versorgt das DVS das IOREG-Register (siehe Makro FCB, Operand IOREG, Seite 430) mit der Adresse des ersten freien Bytes innerhalb des Puffers.
Der Anwender muss anschließend dafür sorgen, dass der Satz, der in die Ausgabedatei aufgenommen werden soll, an dieser Adresse bereitgestellt wird.

Der PUT-Aufruf überprüft den Satz und aktualisiert für den folgenden Satz die Adresse in IOREG. Nachdem der letzte Satz an der IOREG-Adresse bereitgestellt wurde, braucht der PUT-Makro nicht mehr aufgerufen zu werden (andernfalls kommt es zu einem CLOSE-Fehler oder der Datei wird ein „verfälschter“ Satz hinzugefügt).

SAM:

Werden Sätze vom Format V im LOCATE-Modus verarbeitet, gibt das DVS im VARBLD-Register (siehe Makro FCB, Operand VARBLD, Seite 445), nach jedem PUT die verbleibende Pufferkapazität an. Im Anwenderprogramm muss sichergestellt werden, dass ein Satz in den restlichen Pufferbereich vollständig aufgenommen werden kann. Reicht der vorhandene Platz nicht aus, muss der RELSE-Makro aufgerufen werden, damit ein Datenblock abgeschlossen wird.

Move-Mode (Übertragunsbetrieb) – Bandverarbeitung:

Wird bei Bandverarbeitung beim PUT-Aufruf eine Übertragung der vorherigen Sätze ausgelöst und dabei die Bandendemarke erreicht, wird anschließend der aktuelle Satz ohne Blockung allein in einem Block auf das Band geschrieben, bevor der Bandwechsel eingeleitet wird. Dadurch kann sich die Datei pro Bandwechsel um einen Block vergrößern.

Format

Operation	Operanden
PUT	{ fcbadr / (1) }, { area / (0) }[,PARMOD={ 24 / 31 }]

Operandenbeschreibung

fcbadr
Adresse des FCB für die zu verarbeitende Datei.

(1)
Die FCB-Adresse steht im Register 1.

area
aktuelle Adresse des Satzes, der in den Ausgabepuffer gebracht werden soll; bei Dateiverarbeitung im Locate-Mode wird der Operand (Operand IOREG im FCB) ignoriert.

(0)
Die Adresse des in den Ausgabepuffer zu übertragenden Satzes steht im Register 0.

PARMOD
Gibt den Generierungsmodus an.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst. Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum (24-Bit-Adressierung) ablauffähig ist.

= 31
Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist (24-Bit- oder 31-Bit-Adressierung). Der Makroaufruf wird adressierungsmodus-unabängig generiert.

Hinweise zur Programmierung

1. Der PUT-Makroaufruf zerstört die Register 0, 1, 14 und 15.
2. Ein PUT-Makroaufruf führt nur dann zu einem SVC, wenn eine Pufferübertragung ausgelöst wird. Der Anwender kann daher nicht erwarten, bei jedem PUT-Aufruf die Kontrolle in einem STXIT-Prozess zu bekommen (bei Verwendung des STXIT-Makroaufrufs mit SVC= oder SVCLIST=; Näheres siehe Handbuch „Makroaufrufe an den Ablaufteil“ [2]).

PUTX – Satz ersetzen

ISAM:	Makrotyp:	R bei PARMOD=24 O bei PARMOD=31
SAM:	Makrotyp:	R bei PARMOD=24 R bei PARMOD=31

SAM:

Nur für Plattendateien, die im Locate-Mode verarbeitet werden und mit OPEN UPDATE eröffnet wurden: mit dem PUTX-Makroaufruf wird ein aktualisierter Satz in den Puffer zurückgeschrieben. Der Satz muss zuvor mit GET-Makroaufruf (bei ISAM auch GETR, GETKY oder GETFL) bereitgestellt worden sein; bei der Aktualisierung darf die Satzlänge nicht verändert werden.

Im Locate-Mode wird durch einen PUTX-Makroaufruf der Satz in den Programmpuffer gebracht und ein Kennzeichen gesetzt, dass der Pufferinhalt verändert wurde. Erst bei dem nächsten Makroaufruf, der für diese Datei/diesen FCB einen SVC auslöst, wird der Pufferinhalt zurückgeschrieben (z.B. beim nächsten GET).

ISAM:

Der PUTX-Makroaufruf überschreibt einen Satz, der zuvor mit einem GET-, GETR-, GETKY oder GETFL-Makroaufruf gelesen wurde. Bei GETFL muss bei SHARUPD=YES der Operand LOCK gewählt werden.

Der Satzschlüssel darf weder im Move-Mode noch im Locate-Mode verändert werden; im Locate-Mode darf auch die Satzlänge nicht verändert werden.

Zwischen der Leseoperation mit GET und der Schreiboperation mit PUTX darf – außer OSTAT – kein anderer ISAM-Aktionsmakro für diese Datei ausgeführt werden. Bei Shared-Update-Verarbeitung darf die Sperre, die mit dem Leseaufruf explizit oder implizit mit dem Operanden LOCK gesetzt wurde, nicht aufgehoben werden; d.h. auch für eine andere Datei/einen anderen FCB dürfen nur Aktionsmakros ausgeführt werden, die keine neue Sperre setzen, z.B. Leseaufruf mit NOLOCK.

K-ISAM:

Wird für einen Satz, der im Bereich von Sätzen gleicher Schlüssel liegt, im Move-Mode ein PUTX ausgegeben, ist es bei K-ISAM in Verbindung mit SHARUPD=YES unbestimmt, auf welchen Satz mit gleichem Schlüssel nach dem PUTX positioniert wird.

Zwischen Lese- und Schreiboperation darf der Satzschlüssel nicht geändert werden. Wird er dennoch verändert, ist die Auswirkung abhängig davon, ob die Datei im Übertragungs- oder im Locate-Mode verarbeitet wird:

In beiden Fällen wird der EXLST-Ausgang USERERR genommen.

Werden im Locate-Mode Steuerinformationen überschrieben, kann dies bei der nachfolgenden Verarbeitung zu unvorhergesehenen Ergebnissen führen. Wenn der Datenblock von ISAM anschließend nicht mehr verarbeitet werden kann, geht die Steuerung an den EXLST-Ausgang USERERR.

Format

Operation	Operanden
PUTX	{ fcbadr / (1) } [,PARMOD={ 24 / 31 }] [,{ area / (0) }]

Operandenbeschreibung

area
Adresse des Satzes, der geschrieben werden soll

(0)
Die Adresse des Satzes steht im Register 0.

fcbadr
Adresse des FCB der zu verarbeitenden Datei.

(1)
Die FCB-Adresse steht im Register 1.

PARMOD
Gibt den Generierungsmodus an.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst. Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (nur im 24-Bit-Adressierungsmodus).

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig (24-Bit- oder 31-Bit-Adressierung) generiert. Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist.

Hinweise zur Programmierung

1. Der PUTX-Makroaufruf zerstört die Register 0, 1, 14 und 15.
2. *Nur für SAM:*
   Die PUTX-Verarbeitung „merkt“ sich im TU-FCB, dass im aktuellen Puffer ein Satz aktualisiert worden ist. Wenn beim nächsten GET (RELSE, SETL) dieses Bit gesetzt ist, dann erst wird der ganze logische Block auf Platte geschrieben. Der PUTX setzt **nie** einen SVC ab.

RDTFT – Informationen aus TFT und TST anfordern

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form) (siehe Seite 870)

Der Benutzer kann sich mit dem RDTFT-Makroaufruf Informationen aus der TFT in einen Benutzerbereich ausgeben lassen. Gegebenenfalls erhält er auch Informationen aus dem mit dem TFT-Eintrag verknüpften TST-Eintrag.

Format

Operation	Operanden
RDTFT	ausadr [,[länge][,{SHORT / LONG}]] [,FILE=pfadname] [,LINK=name] [,NUMONLY={NO / YES}] [,MF=L][,{PARMOD={24 / 31} / [LINKWC={NO / YES},]VERSION={2 / 3}}]
	MF=(E,{adr / (r)})[,{PARMOD={24 / 31} / VERSION={2 / 3}}]
	MF={D / C}[,PREFIX={pre / *}],PLIST={INPUT / OUTPUT}[,{PARMOD={24 / 31} / VERSION={2 / 3}}]

Operandenbeschreibung

ausadr
Symbolische Adresse des Ausgabebereichs, in den die Informationen aus der TFT übertragen werden sollen. Die Angabe dieser Adresse ist obligatorisch bei Angabe von MF=L oder fehlender MF-Angabe.

länge
Länge des Benutzerbereichs
Mindestlänge: 11 Byte; Ausnahme bei NUMONLY=YES: 4 Byte

Voreinstellung: bei SHORT, PARMOD=24/31: 140 Byte
bei SHORT, VERSION=2/3: 180 Byte
bei LONG: 2048 Byte

FILE = pfadname
legt fest, über welche Datei(en) informiert werden soll, mit:
<c-string 1..80: filename 1..54 with-wild(80)>

Bezeichnet die Datei/Dateigeneration, aus deren TFT-Eintrag Informationen ausgegeben werden sollen.
In „catid“, „userid“ und „dateiname“ können Zeichenfolgen durch Muster ersetzt werden. In diesem Fall darf die Gesamtzeichenfolge für „pfadname“ bis zu 80 Zeichen lang sein. Leere Dateinamen und der Dateiname „*DUMMY“ werden bei Verwendung von Wildcards nicht selektiert.

Ist „pfadname“ eine Dateigeneration, muss die absolute Generationsnummer angegeben werden.

Bei temporären Dateien wird der interne Dateiname ausgegeben.

Pfadname bedeutet [:catid:][$userid.]dateiname

catid
Katalogkennung der Datei; fehlt die Angabe, wird die Default-Catid zugewiesen, die der Userid zugeordnet ist.

userid
Benutzerkennung der Datei; fehlt der Operand, wird die eigene Benutzerkennung angenommen.

dateiname
teil- oder vollqualifizierter Dateiname.

LINK = name
Gibt den Dateikettungsnamen des TFT-Eintrags an, aus dem Informationen in den Ausgabebereich übertragen werden (ggf. zusammen mit Informationen aus dem zugehörigen TST-Eintrag).
In „name“ können Zeichenfolgen durch Muster ersetzt werden. In diesem Fall gilt:
– Die Gesamtzeichenfolge darf bis zu 80 Zeichen lang sein.
– Die Gesamtzeichenfolge darf außer den Wildcards nur Zeichen aus dem zulässigen Wertebereich der Kommandoschnittstelle enthalten.
– Selektiert werden bei der Angabe von Wildcards nur solche TFT-Einträge, deren Dateikettungsname aus Zeichen aus dem zulässigen Wertebereich der Kommandoschnittstelle gebildet wurde.

LINKWC
Nur ab VERSION=2:
legt fest, ob Platzhalterzeichen im Kettungsnamen als Muster oder als gewöhnliche Zeichen interpretiert werden.

= NO
Platzhalterzeichen werden als gewöhnliche Zeichen aufgefasst.

= YES
Platzhalterzeichen werden als Muster aufgefasst.

LONG
Der vollständige TFT-Eintrag und daran anschließend der damit verknüpfte TST-Eintrag einschließlich der Geräteliste werden in den Benutzerbereich übertragen.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben. In allen Makroaufrufen, die sich durch den MF-Operanden unterscheiden (MF=L/E/D/C), muss der Versionsoperand den gleichen Wert haben.
Ist MF nicht angegeben, erhält man die S-Form; die explizite Angabe MF=S ist jedoch nur für PARMOD=24 möglich.

NUMONLY
Nur ab VERSION=3:
legt fest, ob lediglich die Anzahl der selektierten TFT-Einträge in den Ausgabebereich geschrieben werden soll.

= NO
Die Anzahl der TFT-Einträge wird nicht in den Ausgabebereich übertragen.

= YES
Die selektierten TFT-Einträge werden gezählt und das Ergebnis in die ersten vier Byte des Ausgabebereichs geschrieben. Eine weitere Ausgabe erfolgt nicht. Die Mindestgröße des Ausgabebereichs beträgt 4 Byte.

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an. Werden PARMOD und VERSION zugleich angegeben, wird der PARMOD-Operand ignoriert und eine MNOTE-Meldung erzeugt.

Voreinstellung: 31, falls VERSION angegeben ist,
andernfalls der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst.
Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

PLIST
erzeugt eine Liste mit symbolischen Adressen für den Ein- oder Ausgabebereich, abhängig von den Operanden VERSION und PARMOD.

= INPUT
Liste für den Eingabebereich

= OUTPUT
Liste für den Ausgabebereich

PREFIX
wird nur in Verbindung mit MF=C oder MF=D ausgewertet und legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: PREFIX = I

= pre
Das erste Zeichen der generierten Namen wird durch „pre“ ersetzt;
für den Eingabebereich: 1-2 Zeichen, erstes Zeichen ein Buchstabe;
für den Ausgabebereich: 1 Buchstabe

= *
Es wird kein Präfix generiert.

SHORT
Der statische Teil des TFT-Eintrags und daran anschließend der statische Teil des damit verknüpften TST-Eintrags werden in den Benutzerbereich übertragen (ohne Geräteliste) (siehe Operand LONG).

VERSION
Steuert die Generierung von Operandenliste, SVC und Ausgabebereich. Ein gleichzeitig angegebener PARMOD-Wert wird ignoriert, dabei wird allerdings eine MNOTE-Meldung erzeugt.

Der Returncode wird ausschließlich im Standardheader der Parameterliste abgelegt. Außerdem liefert die RDTFT-Funktion die neue Ausgabeliste (siehe „Hinweise zur Programmierung“ auf Seite 779).

Voreinstellung: Die Generierung der Operandenliste wird durch den Operanden PARMOD gesteuert.

= 2
erzeugt die ab BS2000 V9.5 gültige Operandenliste. Der Returncode wird im Standardheader und in Register 15 abgelegt.

= 3
bezeichnet die Makroversion: es wird die Operandenliste für die ab BS2000/OSD-BC V3.0 gültige Makroversion generiert.

Hinweise zur Programmierung

Eingabebereich

Ohne den Operanden VERSION wird die bisherige Operandenliste generiert – abhängig vom Operanden PARMOD. Eine DSECT für diesen Bereich kann generiert werden mit dem Makroaufruf DMARD oder mit den Operanden MF=D,PLIST=INPUT des RDTFT-Makroaufrufs.

Bei Angabe von VERSION=2/3 enthält die Operandenliste den Standardheader. Außerdem ist das Feld zur Aufnahme der Länge des Ausgabebereichs als Adresskonstante (4 Byte) definiert. Eine DSECT für diesen Bereich wird mit den Operanden VERSION=2/3, MF=D, PLIST=INPUT erzeugt.

Ausgabebereich

Der RDTFT-Makroaufruf stellt – je nach Verwendung des LINK-Operanden – zwei Ausgabelisten zur Verfügung:

– ohne LINK=name oder falls „name“ Wildcards enhält und der Operand LINKWC=YES gesetzt ist, wird nur eine Liste der Dateikettungsnamen und der damit verknüpften Dateinamen in den Ausgabebereich übertragen.

  Die Liste wird in chronologischer Reihenfolge ausgegeben; d.h. der Dateikettungsname des zuerst erstellten TFT-Eintrags wird an erster Stelle übertragen. Jedem Paar LINK-Name/Dateiname wird ein Byte vorangestellt, das die Länge dieser beiden Felder + 1 angibt. Die Liste wird abgeschlossen mit einem Byte, das den Wert X'00' enthält. Das Folgebyte zeigt an, ob alle Informationen in den Benutzerbereich übertragen werden konnten:

  X'00' Alle Dateikettungsnamen und deren Dateinamen sind in den Benutzerbereich übertragen worden.

  X'01' Ein oder mehrere Dateikettungsnamen, samt Dateinamen, konnten nicht in den bereits gefüllten Benutzerbereich übertragen werden.

  Ist NUMONLY=YES spezifiziert, wird in den ersten vier Byte des Ausgabebereichs die Anzahl der selektierten TFT-Einträge ausgegeben. Eine weitere Ausgabe erfolgt nicht.

– mit LINK=name werden TFT- und TST-Informationen für den angegebenen Dateikettungsnamen ausgegeben, sofern „name“ keine Wildcards enthält.

Der Ausgabebereich gliedert sich dann in einen festen und einen variablen Teil: der feste Teil enthält Informationen aus TFT und TST, der variable Teil wird nur aufgebaut, wenn der Operand LONG angegeben wurde; er enthält weitere Informationen aus der TFT und evtl. TST-Datenträger-Informationen.

Feldlänge (Byte)		**Inhalt**
mit VERSION	**ohne VERSION**	
2	2	Länge des Ausgabebereichs ohne variablen Teil
8	8	Dateikettungsname
54	54	Pfadname
116	76	statischer Teil des TFT-Eintrags und des damit verbundenen TST-Eintrags
variabel	variabel	TFT- und TST-Datenträgerinformationen (bei Angabe von LONG)

Ohne den Operanden VERSION wird die Ausgabeliste im bisherigen Format erzeugt. Mit dem Makroaufruf DMADR oder den Operanden MF=D, PLIST=OUTPUT des RDTFT-Makroaufrufs kann eine DSECT für diese Ausgabeliste erzeugt werden.

Mit dem Operanden VERSION=2/3 wird die neue Ausgabeliste erzeugt, die gegenüber der bisherigen um einige Felder erweitert wurde. Insbesondere stellt die neue Ausgabeliste zusätzlich zur bisherigen Informationen über BLKCTRL, POOLLNK, TAPEWR, CLOSMSG, CLOSE, IOPERF, IOUSAGE, EXC32GB und DESTOC (siehe FILE-Makro, Seite 455) zur Verfügung. Außerdem wird der Gerätetyp in abdruckbarer Form (z.B. „D3439-10“) ausgegeben. Für die Ausgabe mit LONG wurde die TFT-Datenträger-Information erweitert. Eine DSECT für die Ausgabeliste kann mit den Operanden VERSION=2/3, MF=D, PLIST=OUTPUT erzeugt werden.

– Bei PARMOD=31 oder VERSION=2/3 wird der Pfadname stets vollständig ausgegeben, andernfalls in der Form, in der er über den FILE- oder OPEN-Makro in den TFT-Eintrag gebracht wurde, d.h. gegebenenfalls ohne Katalog- oder Benutzerkennung. Systemspezifisch kann jedoch festgelegt werden, dass Katalog- und Benutzerkennung immer ausgegeben werden.

  Ist der TFT-Eintrag nicht mit einem TST-Eintrag verknüpft, wird der entsprechende Teil des Ausgabebereiches auf binär null (X'00') gesetzt.

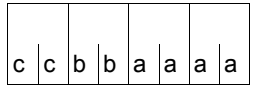
Returncodes

Der Fehlercode wird im Standardheader des Parameterbereichs zurückgegeben. Fehlercode 0 bedeutet, dass kein Fehler aufgetreten ist. Die anderen Fehlercodes sind im Makro DMAIDEM bzw. DCOIDEM beschrieben.

Eine Programm-Terminierung mit STXIT-Anschluss kann in folgenden Fällen eingeleitet werden:
- Parameteradresse fehlerhaft (z.B. kürzer als der Standardheader)
- Parameteradresse nicht wort-ausgerichtet
- UNIT oder FUNCTION im Header fehlerhaft
- Header nicht beschreibbar

Ist die Versionsangabe fehlerhaft, wird der Returncode im Header und in Register 15 zurückgeliefert. Ist der Header nicht beschreibbar, erfolgt Programm-Terminierung mit STXIT-Anschluss. Die Subcodes werden nur ab VERSION=3 versorgt.

Standard-header:



Über die Ausführung des Makros RDTFT wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
	X'01'	X'059D'	Ungültiger Linkname
	X'01'	X'05AB'	Ungültiger Bereich oder Länge
	X'40'	X'05E1'	Linkname nicht gefunden
	X'01'	X'06CB'	Ausgabebereich zu klein
	X'01'	X'06FD'	Ungültiger Parameterlisten-Bereich
	X'03'	X'FFFF'	Ungültige Version

Beispiel

```
BEGIN   START
        .
        .
        RDTFT MF=(E,BEREICH),VERSION=3                  RDTFT-SVC
        .
        .
BEREICH RDTFT AUSGABE,,SHORT,LINK=OTTO,MF=L,VERSION=3   OPERANDENLISTE
        .
        .
AUSGABE DS    XL180                                     AUSGABEBEREICH
        .
EINPL   RDTFT MF=D,PLIST=INPUT,VERSION=3      DSECT EINGABE-OPER-LISTE
        .
AUSPL   RDTFT MF=D,PLIST=OUTPUT,VERSION=3     DSECT AUSGABE-OPER-LISTE
        .
        .
        END
```

RELSE – Block abschließen

Makrotyp: R-Typ

Der RELSE-Makroaufruf bewirkt

– für Dateien, die mit OPEN INPUT/REVERSE/UPDATE eröffnet wurden, dass beim nächsten GET-Makroaufruf die im Puffer verbliebenen Sätze übergangen werden und der 1. Satz im nächsten Datenblock gelesen wird.

– für Dateien, die mit OPEN OUTPUT/EXTEND eröffnet wurden, dass beim nächsten PUT-Makroaufruf der Datenblock auf den Datenträger geschrieben wird und der nächste Satz zum ersten Satz des neuen Datenblocks wird.

  Im Locate-Mode (Ortungsbetrieb) versorgt das DVS das IOREG-Register (siehe Makro FCB, Operand IOREG, Seite 430) mit der Adresse des ersten freien Bytes innerhalb des Puffers. Der Anwender muss anschließend dafür sorgen, dass der Satz, der in die Ausgabedatei aufgenommen werden soll, an dieser Adresse bereitgestellt wird.

  Werden Sätze vom Format V im Locate-Mode verarbeitet, so wird im VARBLD-Register (siehe Makro FCB, Operand VARBLD, Seite 445) die freie Pufferkapazität angezeigt. Nach dem RELSE-Aufruf beträgt die freie Pufferkapazität: Blocklänge minus Pufferverschiebung (siehe BLKSIZE bzw. BUFOFF im Makro FCB, Seite 420 bzw. Seite 423).

– für Dateien, die mit OPEN UPDATE eröffnet wurden, dass der aktuelle Datenblock zurückgeschrieben wird – wenn ein Satz mit PUTX aktualisiert worden ist – und der nächste GET den ersten Satz des nächsten Datenblocks liefert.

Format

Operation	Operanden
RELSE	{ fcbadr / (1) } [,PARMOD={ 24 / 31 }][,SYNC={ NO / YES }]

Operandenbeschreibung

fcbadr
Adresse des FCB, mit dem die zu verarbeitende Datei in Verbindung steht.

(1)
Die FCB-Adresse steht im Register 1.

PARMOD
Gibt den Generierungsmodus für den Makroaufruf an.

Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst.
Das Objekt ist nur im 24-Bit-Adressierungsmodus ablauffähig.

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig generiert.

SYNC
Die Verarbeitung von Ausgabedateien kann synchronisiert werden.

Voreinstellung: SYNC=NO

= NO
Die Dateiverarbeitung wird nicht synchronisiert.

= YES
nur in Verbindung mit PARMOD=31 möglich; die Dateiverarbeitung wird synchronisiert, d.h. nach einem RELSE..., SYNC=YES sind alle Daten auf Platte oder Band geschrieben, es steht kein WAIT mehr aus.

Bei Dateien auf Magnetbandkassetten bewirkt RELSE..., SYNC=YES, dass der Gerätepuffer geleert wird und die Daten auf die Magnetbandkassette geschrieben werden.

Hinweise zur Programmierung

1. Der RELSE-Makroaufruf zerstört die Register 0, 1, 14 und 15.
2. Ein RELSE-Makroaufruf führt nur dann zu einem SVC, wenn eine Pufferübertragung ausgelöst wird. Der Anwender kann daher nicht erwarten, bei jedem RELSE-Aufruf die Kontrolle in einem STXIT-Prozess zu bekommen (bei Verwendung des STXIT-Makroaufrufs mit SVC= oder SVCLIST=; Näheres siehe Handbuch „Makroaufrufe an den Ablaufteil“ [2]).

RELTFT – TFT-Eintrag löschen

Makrotyp: S-Typ (C-Form/D-Form/E-Form/L-Form/M-Form) (siehe Seite 870)

Der RELTFT-Makroaufruf löscht in der Task File Table (TFT) den Eintrag, dessen Dateikettungsname angegeben wird, und gibt alle privaten Datenträger, Geräte und Net-Storage-Volumes, die damit verknüpft waren, frei. Wenn ein privater Datenträger oder ein Net-Storage-Volume mit mehreren TFT-Einträgen verbunden ist, dann wird er erst freigegeben, wenn der letzte TFT-Eintrag gelöscht wird. RELTFT hebt auch die Reservierung von Dateien auf, die mit einem SECURE-RESOURCE-ALLOCATION-Kommando exklusiv reserviert wurden. Er wird ignoriert, wenn für den TFT-Eintrag noch eine LOCK-FILE-LINK-Sperre besteht, und erst dann ausgeführt, wenn diese Sperre mit dem DROPTFT-Makro (siehe Seite 310) oder dem UNLOCK-FILE-LINK-Kommando aufgehoben wird (oder bei LOGOFF; näheres zu den Kommandos LOCK-FILE-LINK, UNLOCK-FILE-LINK und SECURE-RESOURCE-ALLOCATION in den Handbüchern „Kommandos“ [3] und „DVS-Einführung“ [1]).

Hinweis

Der RELTFT-Makroaufruf ist eine Erweiterung des bisherigen REL um die Verwendung von Wildcards im Linknamen. Die bisherige Funktionalität von REL wird unterstützt. Das Format des REL-Makros wird deshalb im Anhang noch gezeigt (siehe Seite 901). Die Operanden des REL-Makros entsprechen jedoch Operanden des RELTFT-Makros. Sie sind deshalb nur hier beschrieben.

Banddateien

Der Anwender kann im RELTFT-Makroaufruf wählen, ob die für die Banddatei angeforderten Bandgeräte (Operand KEEP) und/oder die angeforderten Datenträger (Operand UNLOAD) dem Auftrag zugeordnet bleiben.

Ist der freizugebende TFT-Eintrag mit einem TST-Eintrag verknüpft, wird der Dateizähler im TST-Eintrag um 1 vermindert. Sobald er den Wert 0 erreicht, wird der TST-Eintrag gelöscht, und das DVS gibt alle mit diesem TST-Eintrag verbundenen Geräte frei. Solange der TST-Eintrag noch mit TFT-Einträgen verbunden ist (Dateizähler > 0), gibt das DVS nur die Geräte frei, die nur für den im RELTFT-Makroaufruf genannten TFT-Eintrag angefordert wurden.

Verweist der freizugebende TFT-Eintrag nicht auf einen TST-Eintrag, werden alle mit dem TFT-Eintrag verbundenen Geräte freigegeben.

Format

Operation	Operanden
RELTFT	KEEP = *NO / *YES ,LINK = <c-string 1..80: filename 1..8 with-wild(80) without-gen> / <var: char:80: filename 1..8 with-wild(80) without-gen> ,UNLOAD = *NO / *YES ,VERSION = <integer 1..1> ,WILDCRD = *NO / *YES ,MF = C / D / E / L / M ,PARAM = <adr> / <(r)> ,PREFIX = D / <pre> ,MACID = MAR / <macid>

Operandenbeschreibung

KEEP
Legt fest, ob Bandgeräte, die mit dieser Datei bzw. diesem TFT-Eintrag verbunden sind, nicht an das System zurückgegeben werden sollen, sondern dem Auftrag zur Neuzuweisung zur Verfügung stehen.

= *NO
Die Bandgeräte werden zurückgegeben.

= *YES
Die Bandgeräte werden nicht zurückgegeben.

LINK
Dateikettungsname (Linkname) des zu löschenden TFT-Eintrags.

Im angegebenen Namen können Zeichenfolgen durch Muster ersetzt werden. In diesem Fall gilt Folgendes:

1. Die Gesamtzeichenfolge des Namens darf bis zu 80 Zeichen lang sein.
2. Die Gesamtzeichenfolge des Namens darf außer Wildcards nur Zeichen aus dem zulässigen Wertebereich der Kommandoschnittstelle enthalten.
3. Bei der Angabe von Wildcards werden nur solche TFT-Einträge selektiert, deren Kettungsname aus Zeichen aus dem zulässigen Wertebereich der Kommandoschnittstelle gebildet wurde.

Voreinstellung: Der erste TFT-Eintrag mit dem Linknamen *BLANK wird gelöscht.

= <c-string 1..80: filename 1..8 with-wild(80) without-gen>
Dateikettungsname (Angabe in Hochkommas)

= <var: char: 80: filename 1..8 with-wild(80) without-gen>
Name einer Variablen, die den Dateikettungsnamen enthält

MACID
wird nur in Verbindung mit MF=C/D/M ausgewertet und legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: MACID = MAR

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

Voreinstellung: Operandenliste und SVC wie bisher

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

= adr
adr ist die symbolische Adresse (der Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

PREFIX
wird nur in Verbindung mit MF=C/D/M ausgewertet und legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein Zeichen langes Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen sollen.

UNLOAD
Nur für Banddateien:
Gibt an, ob die mit dem unter LINK angegebenen TFT-Eintrag verbundenen Datenträger freigegeben und die entsprechenden Bandgeräte entladen werden. Soll der Auftrag nochmals auf diese Datenträger zugreifen, müssen sie erneut angefordert werden, sie können nicht mehr automatisch durch das System zugewiesen werden.

Ist ein privater Datenträger mit verschiedenen TFT-Einträgen verbunden, wird er erst dann freigegeben, wenn der letzte TFT-Eintrag gelöscht wird.

= *NO
Die Bandgeräte werden freigegeben.

= *YES
Die Bandgeräte werden nicht freigegeben.

VERSION = <integer 1..1>
Kontrolloperand; steuert Generierung

WILDCRD
Legt fest, ob Platzhalterzeichen in der Kettungsnamenangabe als Muster oder als gewöhnliche Zeichen aufgefasst werden.

= *NO
Platzhalterzeichen werden als gewöhnliche Zeichen aufgefasst.

= *YES
Platzhalterzeichen werden als Muster aufgefasst.

Hinweis zur Programmierung

Der Fehlercode wird nur noch im Standardheader und nicht mehr wie beim REL-Makroaufruf im Mehrzweckregister 15 zurückgeliefert. Eine Programm-Terminierung mit STXIT-Anschluss kann in folgenden Fällen eingeleitet werden:

– Parameteradresse fehlerhaft (z.B. kürzer als der Standardheader)

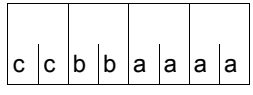
– Parameteradresse nicht wort-ausgerichtet

– UNIT oder FUNCTION im Header fehlerhaft

– Header nicht beschreibbar

Returncodes

Standard-header:



Über die Ausführung des Makros RELTFT wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2, bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'01'	X'00'	X'059A'	Kein derartiger Linkname
	X'01'	X'059D'	Ungültiger Linkname
	X'82'	X'059B'	Datei ist momentan geöffnet
X'02'	X'00'	X'059C'	nicht alle ausgewählten TFT-Einträge gelöscht
	X'01'	X'05C2'	Ungültiger Linkname (binär null)
	X'01'	X'06F5'	TPR-Bit von TU-Aufrufer gesetzt
	X'01'	X'06FD'	Ungültiger Parameterlisten-Bereich
	X'03'	X'FFFF'	Ungültige Version

REMPLNK – Poolkettungsnamen löschen

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form) (siehe Seite 870f)

Der Makro REMPLNK löscht einen oder alle Poolkettungsnamen aus der Pooltabelle des aufrufenden Auftrags. Soll ein Poolkettungsname gelöscht werden, muss die mit ihm verbundene Datei ordnungsgemäß geschlossen worden sein. Gilt der REMPLNK-Makroaufruf für die gesamte Pooltabelle, bleiben die Poolkettungsnamen erhalten, die noch mit einer geöffneten Datei verbunden sind, der Makroaufruf gilt jedoch als erfolgreich ausgeführt.

Format

Operation	Operanden
REMPLNK	MF=L,MODE={ SINGLE ,LINKNAME=name / ALL }
	MF=E,PARAM={ adr / (r) }[]
	MF=D[,PREFIX=pre]
	MF=C[,PREFIX=pre][,MACID=macid]

Operandenbeschreibung

MACID
wird nur in Verbindung mit MF=C ausgewertet und legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: MACID = ISR

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MF
Die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

MODE
gibt an, ob ein bestimmter oder alle Poolkettungsnamen in der Pool-Tabelle des Auftrags gelöscht werden sollen.

= SINGLE
Der Anwender muss einen Poolkettungsnamen angeben. Der in LINKNME angegebene Poolkettungsname soll gelöscht werden.

LINKNME = name
gibt an, welcher Poolkettungsname gelöscht werden soll; „name“ ist ein mit ADDPLNK-Makro definierter Poolkettungsname.

= ALL
Alle Poolkettungsnamen, die nicht mit geöffneten Dateien verbunden sind, sollen gelöscht werden.

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870)

= adr
adr ist die symbolische Adresse (der Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

PREFIX
wird nur in Verbindung mit MF=C oder MF=D ausgewertet und legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein Zeichen langes Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen sollen.

Returncodes

Die mit der C- oder D-Form des Makros generierten Feldnamen und EQU-Anweisungen für die Returncodes beginnen standardmäßig mit der Zeichenfolge DISR, die durch PREFIX und MACID geändert werden kann.

Die Returncodes werden im Standardheader der Parameterliste hinterlegt.

Haupt-Returncode	**Bedeutung**
DISROK X'0000'	Makroaufruf war erfolreich
DISRNPAR X'0001'	auf die Parameterliste kann nicht zugegriffen werden
DISRNREM X'0002'	ISAM-Pools auf remote Host nicht unterstützt
DISRINVN X'0003'	Catid ist nicht bekannt
DISRNACC X'0004'	Catid ist nicht zugreifbar
DISRINVN X'0005'	angegebener Poolkettungsname ist ungültig
DISRNANF X'0006'	Poolkettungsname wurde nicht gefunden
DISRPUSE X'0009'	der angegebene Poolkettungsname ist noch mit einer geöffneten Datei verbunden
DISDSYSE X'000B'	während der Makrobearbeitung trat ein Systemfehler auf
DISRRLNK X'FFFF'	Makroaufruf konnte nicht ausgeführt werden (Linkage Fehler): Sub-Returncode1 auswerten!

RETRY – Makroaufruf wiederholen

Makrotyp: R bei PARMOD=24
0 bei PARMOD=31

Der RETRY-Makroaufruf wird benötigt in Routinen, die bei einem PGLOCK-Ereignis aktiviert werden (EXLST-Ausgang PGLOCK). Das Ereignis PGLOCK tritt nur dann ein, wenn ein ISAM-Aktionsmakroaufruf für einen Satz oder einen Datenblock ausgeführt werden soll, der bereits durch einen anderen Auftrag gesperrt ist. Mit RETRY wird der „erfolglose“ Makroaufruf wiederholt, wobei im Programm definiert werden kann, ob auf das Aufheben der Sperre gewartet werden soll.

Der RETRY-Makro darf nur in PGLOCK-Routinen aufgerufen werden. Wird er an anderer Stelle verwendet, geht die Steuerung an den EXLST-Ausgang USERERR.

Beim Ansprung des PGLOCK-Ausgangs enthält Register 1 die FCB-Adresse; bei der Ausführung des RETRY-Makroaufrufs muss Register 1 wieder diese FCB-Adresse enthalten.

Geht nach erfolgreichem RETRY-Aufruf die Steuerung zurück an den Aufrufer, haben die Register die gleichen Inhalte wie nach einem sofort erfolgreichen Aktionsmakroaufruf.

Ist der RETRY-Aufruf nicht erfolgreich und geht die Steuerung an die FAIL-Routine, haben die Register den gleichen Inhalt wie vor dem RETRY-Aufruf (eine Ausnahme bilden die Register 0, 1, 14 und 15).

Bei NK-ISAM sind die Zeiger vor Ansprung der PGLOCK-Routine so positioniert wie vor dem Makroaufruf (siehe Tabelle „Regeln für ISAM-Zeiger“ auf Seite 79).

Bei K-ISAM ist zu berücksichtigen: wird der PGLOCK-Ausgang des EXLST-Makroaufrufs angesprungen, sind die internen Zeiger nur dann richtig positioniert, wenn der verursachende Makroaufruf ein PUTX oder ELIM (ohne KEY) ist. Die ISAM-Makroaufrufe GET, GETR und GETFL haben den Zeiger bereits verändert, bevor sie nach PGLOCK verzweigen. Der Makroaufruf RETRY führt die Rückpositionierung durch und kann den verursachenden Makroaufruf wieder anstoßen, der den PGLOCK-Ausgang angesprungen hat.

Gehört der Satz, auf den zuletzt erfolgreich zugegriffen wurde, bevor der PGLOCK-Ausgang genommen wurde, zu einer Folge von Sätzen gleicher Schlüssel, wird beim Zurückpositionieren die Datei auf den ersten Satz dieser Folge positioniert.

Wird ACTION=POS angegeben und ist die Positionierung erfolgreich, wird die Steuerung an den Auftrag zurückgegeben, und zwar an den Befehl, der dem RETRY-Makroaufruf folgt.

Format

Operation	Operanden
RETRY	FAIL=adraus[,ACTION={RETRY [,COUNT=zahl] / WAIT / POS}][,PARMOD={24 / 31}]

Operandenbeschreibung

FAIL = adraus
Adresse, an die die Steuerung übergeben werden soll, falls RETRY nicht durchgeführt werden kann oder die Wartezeit für einen Block länger als 30 Minuten ist.

ACTION
gibt an, welche Aktion der RETRY-Makroaufruf ausführen soll.

= RETRY
der Makroaufruf soll wiederholt werden; der Operand COUNT gibt an, wie oft der Makroaufruf wiederholt werden soll, bevor die Steuerung an die FAIL-Adresse geht.

= WAIT
Der Makroaufruf wird wiederholt; ist die Sperre auch dann noch nicht aufgehoben, wird der Auftrag in eine Warteschlange eingereiht. Nach maximal 30 Minuten Wartezeit geht die Steuerung an die FAIL-Routine.

= POS
für K-ISAM: es soll nur der Zeiger in der Datei zurückpositioniert werden.
Bei NK-ISAM wird dieser Wert nur aus Kompatibilitätsgründen unterstützt.

COUNT = zahl
gibt an, wie oft der Makroaufruf wiederholt werden soll, bevor die Steuerung an die FAIL-Routine geht; $0 < zahl \leq 255$
Voreinstellung: COUNT = 1

PARMOD
Gibt den Generierungsmodus an.
Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24
Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (nur 24-Bit-Adressierung).

= 31
Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist (24-Bit- oder 31-Bit-Adressierung).

RFFSNAP– Dateien von einem Snapset restaurieren

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Der Makro RFFSNAP restauriert Dateien eines Pubsets aus einer Pubset-Kopie, die auf einem zugehörigen Snapset erstellt wurde. Beim Restore werden einzelne Dateien von den Snapsets in den laufenden Pubset kopiert. Der Vorgang ist vergleichbar mit einem HSMS-Restore aus einem Backup-Archiv.

Mit der Snapset-Angabe kann ein bestimmter Sicherungsstand (voreingestellt ist die jüngste Snapset-Sicherung) vorgegeben werden oder es soll jede Datei jeweils von dem Snapset mit dem neuesten Dateistand restauriert werden. Vor dem Restaurieren kann sich der Benutzer mit dem Makro LFFSNAP über Dateien informieren, die auf einen Snapset gesichert wurden.

Alle Dateiattribute einer restaurierten Datei werden unverändert von der Originaldatei übernommen (auch Erstellungs- und Änderungsdatum sowie die Schutzattribute). Nur die Allokierung kann von der der Originaldatei abweichen, auch bei Dateien mit physikalischer Allokierung. Dateien auf SM-Pubsets werden auf dem „passendsten“ Volume-Set restauriert. Dieser kann von dem ursprünglichen Volume-Set abweichen.

Einzelne Dateigenerationen können nur mit der gesamten Dateigenerationsgruppe restauriert werden. Dateien auf Privatplatte werden nicht berücksichtigt. Bei migrierten Dateien und Banddateien werden nur die Katalogeinträge restauriert (ohne die Verfügbarkeit der zugehörigen Bänder zu prüfen). Im Falle einer Umbenennung werden diese Dateien ebenfalls nicht berücksichtigt.

Der nichtprivilegierte Benutzer kann die Datei einer fremden Benutzerkennung nur restaurieren, wenn er Miteigentümer ist.

Für bereits vorhandene Dateien muss das Überschreiben durch das Restaurieren explizit zugelassen werden (Operand REPLACE). Für Dateien, die mit Kennwort gegen unberechtigtes Überschreiben geschützt sind, muss das erforderliche Kennwort in der Kennworttabelle des Aufrufers eingetragen sein (Kommando ADD-PASSWORD).

Dateien können auch unter einem neuen Namen restauriert werden. Die Umbenennung erfolgt entweder durch Angabe einer anderen Benutzerkennung (Operand NUSERID) und/oder eines Dateinamenspräfix (Operand NPREFIX).

Optional können Dateien, die zum Zeitpunkt der Snapset-Erzeugung schreibgeöffnet waren, restauriert werden (Operand RESTOPN). Eine so restaurierte Datei hat einen Zustand wie nach einem Systemabsturz. Für eine ISAM-Datei kann ein Verify notwendig werden. Schreibgeöffnete Dateien mit dem Attribut OPNBACK (siehe Makro „CATAL – Katalogeintrag bearbeiten“ auf Seite 131) werden unabhängig von dieser Option restauriert.

Bei Bedarf kann sich der Aufrufer ein Protokoll der Restore-Verarbeitung nach SYSOUT ausgeben lassen (Operand LIST). Das Protokoll kann entweder alle Dateien oder nur die Dateien, die aus bestimmten Gründen nicht restauriert werden konnten, umfassen.

Die Snapsets sind temporär nicht verfügbar, wenn das Subsystem SHC-OSD zum Zeitpunkt des Pubset-Imports noch nicht aktiv war. Der Makroaufruf wird in diesem Fall mit Returncode 0622 abgebrochen. Sobald SHC-OSD aktiv ist, werden die Snapsets bei Aufruf des Kommandos SHOW-SNAPSET-CONFIGURATION nachträglich aktiviert.

Privilegierte Funktionen

Die Systembetreuung (Privileg TSOS) kann als Mit-Eigentümer alle Dateien unter ihren Original-Benutzerkennungen restaurieren.

Beim Überschreiben einer noch bestehenden Datei kann die Systembetreuung den Dateischutz mit dem Operanden IGNPROT explizit umgehen.

Das Restaurieren von Dateien kann die Systembetreuung über die SECOS-Komponente SAT nur protokollieren, wenn sie die intern benutzten Aufrufe zum Dateilöschen (beim Überschreiben) und zum Erstellen eines Eintrags im Dateikatalog protokollieren lässt.

Format

Operation	Operanden
RFFSNAP	,PATHNAM=<c-string 1..80: filename 1..54 with-wild(80)> / <var: char:80> ,SNAPSET=<integer -52..-1> / *LATEST / *ALL ,SNAPID=<c-string 1..1: name 1..1 with-low> / <var: char:1> ,REPLACE=*NO / *YES / <var: enum-of_replace_s: 1> ,IGNPROT=*NO / *YES / <var: enum-of_ignprot_s: 1> ,RESTOPN=*NO / *YES / <var: enum-of_restop_s: 1> ,NUSERID=<c-string 1..8: name 1..8> / <var: char:8> ,NPREFIX=<c-string 1..8: name 1..8> / <var: char:8> ,LIST=*NO / *SYSOUT / *ERRORS-TO-SYSOUT / <var: enum-of_list_s: 1> ,EQUATES=*YES / *NO MF=L
	MF=D,PREFIX=D / <pre>
	MF=E,PARAM=<name 1..27>
	MF=C / M ,PREFIX=D / <pre> ,MACID=MAR / <macid>

PATHNAM
Auswahl der Dateien, die restauriert werden sollen.

=<c-string 1..80: filename 1..54 with-wild(80)>
Pfadname der Datei(en) auf dem Snapset. Mit Musterzeichen kann eine Auswahlangabe für eine Dateimenge erfolgen.

Die Dateien müssen folgende Voraussetzungen erfüllen:
- Sie müssen zum Zeitpunkt der Snapset-Erstellung katalogisiert sein.
- Der Pubset, an dem sie katalogisiert sind, muss lokal importiert sein.
- Sie dürfen nicht auf Privatplatte liegen.

Katalog- und Benutzerkennung müssen eindeutig (also ohne Musterzeichen) angegeben werden. Die Angabe von Aliasnamen (auch teilqualifiziert) ist zulässig. Der Name einer Dateigenerationsgruppe darf angegeben werden, nicht aber der Name einer einzelnen Dateigeneration (einzelne Dateigenerationen können nur innerhalb der Gruppe restauriert werden).

Der privilegierte Benutzer (Privileg TSOS) kann Dateien aller Benutzerkennungen restaurieren.

=<var: char:80>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 80 Byte, in dem der Pfadname bzw. die Musterzeichenfolge für die gewünschte(n) Datei(en) abgelegt ist.

SNAPSET
Der Operand darf nicht zusammen mit dem Operanden SNAPID angegeben werden.
Bezeichnet den Snapset, von dem restauriert werden soll, über das relative Alter.

=<integer -52..-1>
Bezeichnet den Snapset explizit über das relative Alter. Der Wert -1 entspricht dem jüngsten Snapset (entspricht auch *LATEST).

=*LATEST
Bezeichnet den jüngsten Snapset.

=*ALL
Für die Restaurierung werden alle Snapsets des entsprechenden Pubsets als Basis herangezogen. Jede Datei wird jeweils von dem Snapset mit dem neuesten Stand dieser Datei (also mit der letzten Sicherung) restauriert. Eine Datei, die nicht mit dem neuesten Dateistand restauriert werden kann, ist in diesem Fall nicht restaurierbar (d.h. ältere Sicherungsstände werden ignoriert).

SNAPID
Der Operand darf nicht zusammen mit dem Operanden SNAPSET angegeben werden.
Bezeichnet den Snapset, von dem restauriert werden soll, über die Snapset-Id.

=<c-string 1..1: name 1..1 with-low>
Bezeichnet den Snapset explizit über die Snapset-Id. Die maximal 52 Snapsets zu einem Pubset werden unterschieden durch Snapset-Ids aus den 26 Kleinbuchstaben a bis z und den 26 Großbuchstaben A bis Z.

=<var: char:1>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 1 Byte, in dem die Snapset-Id abgelegt ist.

Hinweis
Wenn weder SNAPSET noch SNAPID angegeben sind, wird der jüngste Snapset verwendet.

REPLACE
Gibt an, ob die zu restaurierenden Dateien bereits existierende Dateien überschreiben dürfen.

=*NO
Bereits existierende Dateien werden nicht überschrieben. Das bedeutet, dass Dateien mit Namen bereits existierender Dateien nicht restauriert werden.

=*YES
Bereits existierende Dateien dürfen von zu restaurierenden Dateien überschrieben werden, soweit die Schutzattribute dies zulassen. Für Dateien, die mit Kennwort gegen unberechtigtes Überschreiben geschützt sind, muss das erforderliche Kennwort in der Kennworttabelle des Aufrufers eingetragen sein (siehe Kommando ADD-PASSWORD).

=<var: enum-of_replace_s: 1>
Name des Feldes mit dem Wert für REPLACE.

IGNPROT
Der Operand steht nur dem privilegierten Benutzer (Privileg TSOS) zur Verfügung.
Gibt an, ob Dateien ohne Beachtung eines bestehenden Schreibschutzes überschrieben werden sollen.

=*NO
Der Schreibschutz wird beachtet.

=*YES
Der Schreibschutz wird ignoriert.

=<var: enum-of_ignprot_s: 1>
Name des Feldes mit dem Wert für IGNPROT.

RESTOPN
Gibt an, ob auch Dateien restauriert werden sollen, die beim Sichern auf den Snapset schreibgeöffnet waren und bei denen das Dateiattribut OPNBACK (siehe Makro „CATAL – Katalogeintrag bearbeiten“ auf Seite 131) nicht gesetzt war.

=*NO
Diese Dateien werden nicht restauriert. Restauriert werden also nur Dateien, die beim Sichern auf den Snapset nicht schreibgeöffnet waren, und Dateien, bei denen das Attribut OPNBACK gesetzt war.

=*YES
Auch diese Dateien werden restauriert. Die Konsistenz entspricht der nach einem System-Crash (Schreibzugriffe in korrekter Reihenfolge). Bei ISAM-Dateien kann ein Verify (Kommando REPAIR-DISK-FILE) notwendig werden.

=<var: enum-of_restop_s: 1>
Name des Feldes mit dem Wert für RESTOPN.

NUSERID
Gibt an, dass die Dateien beim Restaurieren umbenannt und unter der angegebenen Benutzerkennung restauriert werden sollen.
Der Operand darf nicht zusammen mit dem Operanden NPREFIX angegeben werden.

=<c-string 1..8: name 1..8>
Benutzerkennung.

=<var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 8 Byte, in dem die Benutzerkennung abgelegt ist.

NPREFIX=
Gibt an, dass die Dateien beim Restaurieren umbenannt werden und dabei den angegebenen Dateinamenspräfix erhalten sollen.
Der Operand darf nicht zusammen mit dem Operanden NUSERID angegeben werden.

=<c-string 1..8: name 1..8>
Dateinamenspräfix.

=<var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 8 Byte, in dem der Dateinamenspräfix abgelegt ist.

LIST
Gibt an, welche Verarbeitungsergebnisse nach SYSOUT protokolliert werden sollen.

=*NO
Es erfolgt keine Ausgabe nach SYSOUT.

=*SYSOUT
Es werden alle Dateien aufgelistet. Für die nicht restaurierbaren Dateien wird die Ursache jeweils mit einem Meldungsschlüssel angezeigt.

=*ERRORS-TO-SYSOUT
Es werden nur Dateien aufgelistet, die nicht restauriert werden konnten. Die Ursache wird jeweils mit einem Meldungsschlüssel angezeigt.

=<var: enum-of_list_s: 1>
Name des Feldes mit dem Wert für LIST.

EQUATES
Steuerungs-Operand nur für MF=C und MF=D:
Gibt an, ob bei der Expansion des Parameterbereichs auch Equates für die Werte der Felder des Parameterbereichs generiert werden sollen.

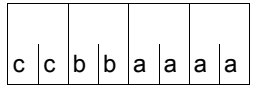
= *YES
Bei der Expansion des Parameterbereichs werden auch Equates für die Werte der Felder des Parameterbereichs generiert.

= *NO
Bei der Expansion des Parameterbereichs werden keine Equates für die Werte der Felder des Parameterbereichs generiert.

Returncodes

Der Returncode wird im Standardheader des Parameterbereichs abgelegt. Der Parameterbereich darf dann nicht im Read-only-Bereich liegen, sonst erfolgt Programmterminierung.

Standard-header:



Über die Ausführung des Makros RFFSNAP wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2,bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'00'	X'0000'	kein Fehler
X'00'	X'40'	X'0501'	angeforderter Katalog nicht verfügbar
X'00'	X'40'	X'0505'	Fehler bei der Rechner-Kommunikation
X'00'	X'40'	X'0512'	angeforderter Katalog nicht gefunden
X'00'	X'40'	X'051B'	gewünschte Benutzerkennung nicht im Pubset
X'00'	X'40'	X'051D'	LOGON-Passwort auf angegebenem Pubset anders
X'00'	X'20'	X'0531'	Unerwarteter Fehler beim Katalogzugriff
X'00'	X'40'	X'0535'	angegebene Datei nicht zugreifbar
X'00'	X'82'	X'053C'	im Katalog des Pubsets ist kein Platz
X'00'	X'82'	X'0541'	nicht genügend Plattenspeicherplatz
X'00'	X'40'	X'0554'	Format des Dateinamens unzulässig
X'00'	X'40'	X'057F'	migrierte Datei kann nicht unbenannt werden
X'00'	X'20'	X'0584'	interner Fehler
X'00'	X'82'	X'0594'	nicht genügend virtueller Speicher
X'00'	X'82'	X'05B1'	für die Datei besteht eine Dateisperre
X'02'	X'00'	X'05B6'	fehlerhafte Zeitkonvertierung GTIME-Makro
X'00'	X'40'	X'05BF'	Kennwort nicht angegeben
X'00'	X'82'	X'05C3'	Datei zurzeit gesperrt oder in Gebrauch
X'00'	X'40'	X'05C6	Freigabedatum noch nicht erreicht
X'00'	X'20'	X'05C7'	interner Fehler im DVS
X'00'	X'01'	X'05CB'	Fehlerhafter erster Dateiname oder fremde Benutzerkennung angegeben
X'00'	X'01'	X'05EE'	Pfadname nach Komplettierung zu lang
X'00'	X'40'	X'05FC'	angegebene Benutzerkennung nicht im Home-Pubset

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'40'	X'0610'	mindestens für einen der ausgewählten Dateinamen liefert die Funktionsausführung einen Returncode
X'00'	X'40'	X'0615'	Datei liegt auf einem nicht verfügbaren Volume Set
X'00'	X'40'	X'0616'	Volume Set in SM-Pubset nicht zugreifbar
X'00'	X'40'	X'0620'	keine restaurierbare Datei gefunden
X'00'	X'40'	X'0621'	Datei bereits katalogisiert, Restaurierung nicht ausgeführt
X'00'	X'40'	X'0622'	Snapset nicht verfügbar
X'00'	X'01'	X'0623'	Generation kann nicht restauriert werden
X'00'	X'01'	X'0624'	Dateiname ungültig
X'00'	X'40'	X'0681'	DVS Fehler bei Zugriff auf Datei
X'00'	X'40'	X'0684'	Datei existiert nicht
X'00'	X'01'	X'06C5'	FGG Name zu lang
X'00'	X'01'	X'06C6'	Name einer Banddatei kann nicht geändert werden
X'00'	X'40'	X'06CC'	kein Dateiname entspricht der angegebenen Musterzeichenfolge
X'00'	X'40'	X'06D5'	Datei geschützt und damit nicht überschreibbar
X'00'	X'01'	X'06F7'	Ungültiger Operandenwert
X'00'	X'01'	X'06FD'	Parameterbereich ungültig oder nicht zugreifbar

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Layout der Operandenliste

Makroauflösung mit MF=D, sowie Standardwerten für EQUATES, PREFIX und MACID:

```
         RFFSNAP MF=D
DMARRFPL DSECT ,
DMARHDR  DS    0A
DMARFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
DMARIFID DS    0A              0   INTERFACE IDENTIFIER
DMARFCTU DS    AL2             0   FUNCTION UNIT NUMBER
DMARFCT  DS    AL1             2   FUNCTION NUMBER
DMARFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
DMARRET  DS    0A              4   GENERAL RETURN CODE
DMARSRET DS    0AL2            4   SUB RETURN CODE
DMARSR2  DS    AL1             4   SUB RETURN CODE 2
DMARSR1  DS    AL1             5   SUB RETURN CODE 1
DMARMRET DS    0AL2            6   MAIN RETURN CODE
DMARMR2  DS    AL1             6   MAIN RETURN CODE 2
DMARMR1  DS    AL1             7   MAIN RETURN CODE 1
DMARFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
DMARPNAM DS    CL80                    PATHNAME
DMARSNAP DS    FL1                     SNAPSET
*   SNAPSET - VALUES
DMARSNIN EQU   0                       SNAPSET=<integer>
DMARSNCH EQU   1                       SNAPSET=<char>
DMARSNLT EQU   2                       SNAPSET=*LATEST
DMARSNAL EQU   3                       SNAPSET=*ALL
*
DMARREPL DS    FL1                     REPLACE
 *   REPLACE - VALUES
DMARREPY EQU   0                       REPLACE = YES
DMARREPN EQU   1                       REPLACE = NO
*
DMARIGNP DS    FL1                     IGNPROT
*   IGNPROT VALUES
DMARIGNO EQU   0                       IGNPROT = NO
DMARIGYE EQU   1                       IGNPROT = YES
*
DMARLIST DS    FL1                     LIST
*   LIST - VALUES
DMARLSTN EQU   0                       LIST = NO
DMARLSYO EQU   1                       LIST = SYSOUT
DMARLSYE EQU   2                       LIST = ERRORS
*
DMARNUSR DS    CL8                     NUSERID
DMARNPRE DS    CL8                     NPREFIX
DMARSNVL DS    H                       SnapValue
DMARSNID DS    CL1                     Snapid
```

```
DMAROFLG DS    AL1                      FLAG BYTE
DMARNUSP EQU   X'80'                    S: NUSERID SPECIFIED
DMARNPSP EQU   X'40'                    S: NPREFIX SPECIFIED
DMARRESB EQU   X'3F'                    RESERVED
DMARRESO DS    FL1                      RESTOPN
*   RESTOPN VALUES
DMARRONO EQU   0                        RESTOPN = NO
DMARROYE EQU   1                        RESTOPN = YES
*
DMARRES1 DS    XL3                      ALIGNMENT
DMAR#    EQU   *-DMARHDR
```

Beispiel für eine Aufruffolge

```
        MVC   RFFSMFC(DMAR#),RFFSMFL
        RFFSNAP MF=M,PATHNAM=':X:TTT',PARAM=RFFSMFC
        RFFSNAP MF=E,PARAM=RFFSMFC
            .
            .
RFFSMFC RFFSNAP MF=C
RFFSMFL RFFSNAP MF=L,...
```

RJFSNAP– Jobvariablen von einem Snapset restaurieren

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Der Makro RJFSNAP restauriert Jobvariablen eines Pubsets aus einer Pubset-Kopie, die auf einem zugehörigen Snapset erstellt wurde. Beim Restore werden einzelne Jobvariablen von den Snapsets in den laufenden Pubset kopiert. Der Vorgang ist vergleichbar mit einem HSMS-Restore aus einem Backup-Archiv.

Mit der Snapset-Angabe kann ein bestimmter Sicherungsstand (voreingestellt ist die jüngste Snapset-Sicherung) vorgegeben werden oder es soll jede Jobvariable jeweils von dem Snapset mit dem neuesten Jobvariablenstand restauriert werden. Vor dem Restaurieren kann sich der Benutzer mit dem Makro LJFSNAP über Jobvariablen informieren, die auf einen Snapset gesichert wurden.

Alle Attribute einer restaurierten Jobvariable werden unverändert von der Originaljobvariable übernommen (auch Erstellungs- und Änderungsdatum sowie die Schutzattribute).

Der nichtprivilegierte Benutzer kann die Jobvariable einer fremden Benutzerkennung nur restaurieren, wenn er Miteigentümer ist.

Für bereits vorhandene Jobvariablen muss das Überschreiben durch das Restaurieren explizit zugelassen werden (Operand REPLACE). Für Jobvariablen, die mit Kennwort gegen unberechtigtes Überschreiben geschützt sind, muss das erforderliche Kennwort in der Kennworttabelle des Aufrufers eingetragen sein (siehe ADD-PASSWORD).

Jobvariablen können auch unter einem neuen Namen restauriert werden (Operand NEW-JV-NAME). Die Umbenennung erfolgt entweder durch Angabe einer anderen Benutzerkennung (Operand NUSERID) und/oder eines Namenspräfix (Operand NPREFIX).

Bei Bedarf kann sich der Aufrufer ein Protokoll der Restore-Verarbeitung nach SYSOUT ausgeben lassen (Operand LIST). Das Protokoll kann entweder alle Jobvariablen oder nur die Jobvariablen, die aus bestimmten Gründen nicht restauriert werden konnten, umfassen.

Die Snapsets sind temporär nicht verfügbar, wenn das Subsystem SHC-OSD zum Zeitpunkt des Pubset-Imports noch nicht aktiv war. Der Makroaufruf wird in diesem Fall mit Returncode 0622 abgebrochen. Sobald SHC-OSD aktiv ist, werden die Snapsets bei Aufruf des Kommandos SHOW-SNAPSET-CONFIGURATION nachträglich aktiviert.

Privilegierte Funktionen

Die Systembetreuung (Privileg TSOS) kann als Mit-Eigentümer alle Jobvariablen unter ihren Original-Benutzerkennungen restaurieren.

Beim Überschreiben einer noch bestehenden Jobvariable kann die Systembetreuung den Schutz mit dem Operanden IGNPROT explizit umgehen.

Das Restaurieren von Jobvariablen kann die Systembetreuung über die SECOS-Komponente SAT nur protokollieren, wenn sie die intern benutzten Aufrufe zum Löschen einer Jobvariablen (beim Überschreiben) und zum Erstellen einer Jobvariablen protokollieren lässt.

Format

Operation	Operanden
RJFSNAP	,JVNAME=<c-string 1..80: filename 1..54 with-wild(80)> / <var: char:80> ,SNAPSET=<integer -52..-1> / *LATEST / *ALL ,SNAPID=<c-string 1..1: name 1..1 with-low> / <var: char 1..1> ,REPLACE=*NO / *YES / <var: enum-of_replace_s: 1> ,IGNPROT=*NO / *YES / <var: enum-of_ignprot_s: 1> ,NUSERID=<c-string 1..8: name 1..8> / <var: char:8> ,NPREFIX=<c-string 1..8: name 1..8> / <var: char:8> ,LIST=*NO / *SYSOUT / *ERRORS-TO-SYSOUT / <var: enum-of_list_s: 1> ,EQUATES=*YES / *NO MF=L
	MF=D,PREFIX=D / <pre>
	MF=E,PARAM=<name 1..27>
	MF=C / M ,PREFIX=D / <pre> ,MACID=MAR / <macid>

JVNAME
Auswahl der Jobvariablen, die restauriert werden sollen.

=<c-string 1..80: filename 1..54 with-wild(80)>
Pfadname der Jobvariable(n) auf dem Snapset. Mit Musterzeichen kann eine Auswahlangabe für eine Jobvariablenmenge erfolgen.

Die Jobvariablen müssen folgende Voraussetzungen erfüllen:
– Sie müssen zum Zeitpunkt der Snapset-Erstellung katalogisiert sein.
– Der Pubset, an dem sie katalogisiert sind, muss lokal importiert sein.

Katalog- und Benutzerkennung müssen eindeutig (also ohne Musterzeichen) angegeben werden. Die Angabe von Aliasnamen (auch teilqualifiziert) ist zulässig.

Der privilegierte Benutzer (Privileg TSOS) kann Jobvariablen aller Benutzerkennungen restaurieren.

=<var: char:80>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 80 Byte, in dem der Pfadname bzw. die Musterzeichenfolge für die gewünschte(n) Jobvariable(n) abgelegt ist.

SNAPSET
Der Operand darf nicht zusammen mit dem Operanden SNAPID angegeben werden.
Bezeichnet den Snapset, von dem restauriert werden soll, über das relative Alter.

=<integer -52..-1>
Bezeichnet den Snapset explizit über das relative Alter. Der Wert -1 entspricht dem jüngsten Snapset (entspricht auch *LATEST).

=*LATEST
Bezeichnet den jüngsten Snapset.

=*ALL
Für die Restaurierung werden alle Snapsets des entsprechenden Pubsets als Basis herangezogen. Jede Jobvariable wird jeweils von dem Snapset mit dem neuesten Stand dieser Jobvariablen (also mit der letzten Sicherung) restauriert. Eine Jobvariable, die nicht mit dem neuesten Stand restauriert werden kann, ist in diesem Fall nicht restaurierbar (d.h. ältere Sicherungsstände werden ignoriert).

SNAPID
Der Operand darf nicht zusammen mit dem Operanden SNAPSET angegeben werden.
Bezeichnet den Snapset, von dem restauriert werden soll, über die Snapset-Id.

=<c-string 1..1: name 1..1 with-low>
Bezeichnet den Snapset explizit über die Snapset-Id. Die maximal 52 Snapsets zu einem Pubset werden unterschieden durch Snapset-Ids aus den 26 Kleinbuchstaben a bis z und den 26 Großbuchstaben A bis Z.

=<var: char 1..1>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 1 Byte, in dem die Snapset-Id abgelegt ist.

Hinweis
Wenn weder SNAPSET noch SNAPID angegeben sind, wird der jüngste Snapset verwendet.

REPLACE
Gibt an, ob die zu restaurierenden Jobvariablen bereits existierende Jobvariablen überschreiben dürfen.

=*NO
Bereits existierende Jobvariablen werden nicht überschrieben. Das bedeutet, dass Jobvariablen mit Namen bereits existierender Jobvariablen nicht restauriert werden.

=*YES
Bereits existierende Jobvariablen dürfen von zu restaurierenden Jobvariablen überschrieben werden, soweit die Schutzattribute dies zulassen. Für Jobvariable, die mit Kennwort gegen unberechtigtes Überschreiben geschützt sind, muss das erforderliche Kennwort in der Kennworttabelle des Aufrufers eingetragen sein (siehe Kommando ADD-PASSWORD).

=<var: enum-of_replace_s: 1>
Name des Feldes mit dem Wert für REPLACE.

IGNPROT
Der Operand steht nur dem privilegierten Benutzer (Privileg TSOS) zur Verfügung.
Gibt an, ob Jobvariablen ohne Beachtung eines bestehenden Schreibschutzes überschrieben werden sollen.

=*NO
Der Schreibschutz wird beachtet.

=*YES
Der Schreibschutz wird ignoriert.

=<var: enum-of_ignprot_s: 1>
Name des Feldes mit dem Wert für IGNPROT.

NUSERID
Gibt an, dass die Jobvariablen beim Restaurieren umbenannt und unter der angegebenen Benutzerkennung restauriert werden sollen.
Der Operand darf nicht zusammen mit dem Operanden NPREFIX angegeben werden.

=<c-string 1..8: name 1..8>
Benutzerkennung.

=<var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 8 Byte, in dem die Benutzerkennung abgelegt ist.

NPREFIX=
Gibt an, dass die Jobvariablen beim Restaurieren umbenannt werden und dabei den angegebenen Namenspräfix erhalten sollen.
Der Operand darf nicht zusammen mit dem Operanden NUSERID angegeben werden.

=<c-string 1..8: name 1..8>
Namenspräfix.

=<var: char:8>
Nur mit MF=M möglich:
Symbolische Adresse eines Speicherbereichs von 8 Byte, in dem der Namenspräfix abgelegt ist.

LIST
Gibt an, welche Verarbeitungsergebnisse nach SYSOUT protokolliert werden sollen.

=*NO
Es erfolgt keine Ausgabe nach SYSOUT.

=*SYSOUT
Es werden alle Jobvariablen aufgelistet. Für die nicht restaurierbaren Jobvariablen wird die Ursache jeweils mit einem Meldungsschlüssel angezeigt.

=*ERRORS-TO-SYSOUT
Es werden nur Jobvariablen aufgelistet, die nicht restauriert werden konnten. Die Ursache wird jeweils mit einem Meldungsschlüssel angezeigt.

=<var: enum-of_list_s: 1>
Name des Feldes mit dem Wert für LIST.

EQUATES
Steuerungs-Operand nur für MF=C und MF=D:
Gibt an, ob bei der Expansion des Parameterbereichs auch Equates für die Werte der Felder des Parameterbereichs generiert werden sollen.

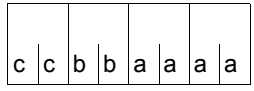
= *YES
Bei der Expansion des Parameterbereichs werden auch Equates für die Werte der Felder des Parameterbereichs generiert.

= *NO
Bei der Expansion des Parameterbereichs werden keine Equates für die Werte der Felder des Parameterbereichs generiert.

Returncodes

Der Returncode wird im Standardheader des Parameterbereichs abgelegt. Der Parameterbereich darf dann nicht im Read-only-Bereich liegen, sonst erfolgt Programmterminierung.

Standard-header:



Über die Ausführung des Makros RJFSNAP wird im Standardheader folgender Returncode übergeben (cc = SUBCODE2,bb = SUBCODE1, aaaa = MAINCODE):

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'00'	X'0000'	kein Fehler
X'00'	X'40'	X'0433'	JV nicht katalogisiert
X'00'	X'40'	X'0435'	JV nicht zugreifbar
X'00'	X'40'	X'0440'	JV Name unzulässig
X'00'	X'40'	X'04A0'	JV Subsystem nicht zugreifbar
X'00'	X'40'	X'04B1'	Kennwort nicht angegeben
X'00'	X'40'	X'04B6'	Freigabedatum noch nicht erreicht
X'00'	X'40'	X'04B8'	nur Lesezugriff erlaubt
X'00'	X'40'	X'04BF'	Zugriff wegen JV-Schutz nicht erlaubt
X'00'	X'40'	X'0501'	angeforderter Katalog nicht verfügbar
X'00'	X'40'	X'0505'	Fehler bei der Rechner-Kommunikation
X'00'	X'40'	X'0512'	angeforderter Katalog nicht gefunden
X'00'	X'40'	X'051B'	gewünschte Benutzerkennung nicht im Pubset
X'00'	X'40'	X'051D'	LOGON-Passwort auf angegebenem Pubset anders
X'00'	X'20'	X'0531'	Unerwarteter Fehler beim Katalogzugriff
X'00'	X'82'	X'053C'	im Katalog des Pubsets ist kein Platz
X'00'	X'20'	X'0584'	interner Fehler
X'00'	X'82'	X'0594'	nicht genügend virtueller Speicher
X'02'	X'00'	X'05B6'	fehlerhafte Zeitkonvertierung GTIME-Makro
X'00'	X'20'	X'05C7'	interner Fehler im DVS
X'00'	X'01'	X'05EE'	Pfadname nach Komplettierung zu lang
X'00'	X'40'	X'05FC'	angegebene Benutzerkennung nicht im Home-Pubset
X'00'	X'40'	X'0610'	mindestens für einen der ausgewählten J- Namen liefert die Funktionsausführung einen Returncode
X'00'	X'40'	X'0620'	keine restaurierbare JV gefunden

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'40'	X'0621'	JV bereits katalogisiert, Restaurierung nicht ausgeführt
X'00'	X'40'	X'0622'	Snapset nicht verfügbar
X'00'	X'01'	X'0624'	JV Name ungültig
X'00'	X'40'	X'0682'	JV Fehler beim Zugriff auf JV
X'00'	X'01'	X'06F7'	Ungültiger Operandenwert
X'00'	X'01'	X'06FD'	Parameterbereich ungültig oder nicht zugreifbar

Die Returncodes mit dem Maincode X'04xy' gehören zur Komponente JVS. Eine Liste mit den Erläuterungen kann mit dem Makro JVSERROR ausgegeben werden (siehe auch Handbuch „Jobvariablen“ [21]).

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Layout der Operandenliste

Makroauflösung mit MF=D, sowie Standardwerten für EQUATES, PREFIX und MACID:

```
         RJFSNAP MF=D
*   PARAMETER AREA
DMAKHDR  FHDR  MF=(C,DMAK),EQUATES=NO
DMAKHDR  DS    0A
DMAKFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
DMAKIFID DS    0A              0   INTERFACE IDENTIFIER
DMAKFCTU DS    AL2             0   FUNCTION UNIT NUMBER
DMAKFCT  DS    AL1             2   FUNCTION NUMBER
DMAKFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
DMAKRET  DS    0A              4   GENERAL RETURN CODE
DMAKSRET DS    0AL2            4   SUB RETURN CODE
DMAKSR2  DS    AL1             4   SUB RETURN CODE 2
DMAKSR1  DS    AL1             5   SUB RETURN CODE 1
DMAKMRET DS    0AL2            6   MAIN RETURN CODE
DMAKMR2  DS    AL1             6   MAIN RETURN CODE 2
DMAKMR1  DS    AL1             7   MAIN RETURN CODE 1
DMAKFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
```

```
*
DMAKJNAM DS    CL80                    JVNAME
DMAKSNAP DS    FL1                     SNAPSET
*   SNAPSET - VALUES
DMAKSNIN EQU   0                       SNAPSET=<integer>
DMAKSNCH EQU   1                       SNAPSET=<char>
DMAKSNLT EQU   2                       SNAPSET=*LATEST
DMAKSNAL EQU   3                       SNAPSET=*ALL
*
DMAKREPL DS    FL1                     REPLACE
*   REPLACE - VALUES
DMAKREPY EQU   0                       REPLACE = YES
DMAKREPN EQU   1                       REPLACE = NO
*
DMAKIGNP  DS    FL1                    IGNPROT
*   IGNPROT VALUES
DMAKIGNO  EQU   0                      IGNPROT = NO
DMAKIGYE  EQU   1                      IGNPROT = YES
*
DMAKLIST  DS    FL1                    LIST
*   LIST - VALUES
DMAKLSTN  EQU   0                      LIST = NO
DMAKLSYO  EQU   1                      LIST = SYSOUT
DMAKLSYE  EQU   2                      LIST = ERRORS
*
DMAKNUSR  DS    CL8                    NUSERID
DMAKNPRE  DS    CL8                    NPREFIX
DMAKSNVL  DS    H                      SnapValue
DMAKSNID  DS    CL1                    Snapid
DMAKOFLG  DS    AL1                    FLAG BYTE
DMAKNUSP  EQU   X'80'                  S: NUSERID SPECIFIED
DMAKNPSP  EQU   X'40'                  S: NPREFIX SPECIFIED
DMAKRES1  EQU   X'3F'                  RESERVED
DMAK#     EQU   *-DMAKHDR
```

Beispiel für eine Aufruffolge

```
          MVC   RJFSMFC(DMAK#),RJFSMFL
          RJFSNAP MF=M,PATHNAM=':X:JV1',PARAM=RJFSMFC
          RJFSNAP MF=E,PARAM=RJFSMFC
              .
              .
 RJFSMFC  RJFSNAP MF=C
 RJFSMFL  RJFSNAP MF=L,...
```

SETL – In der Datei positionieren

ISAM:	Makrotyp:	R bei PARMOD=24 0 bei PARMOD=31
SAM:	Makrotyp:	R bei PARMOD=24 0 bei PARMOD=31

ISAM:

Mit dem SETL-Makroaufruf kann auf den Anfang oder das Ende der Datei positioniert werden oder über die Schlüsselangabe auf jeden beliebigen Satz.

Wird mit SETL KEY in der Datei über den Primärschlüssel positioniert und enthält die Datei Sätze mit gleichen Primärschlüsselwerten (DUPEKY=YES), so wird auf den Ersten dieser Sätze positioniert.

Wird mit SETL KEY in der Datei über einen Sekundärschlüssel positioniert und enthält die Datei Sätze mit gleichen Werten für diesen Sekundärschlüssel, so wird auf den Satz positioniert, auf dessen Primärschlüssel im zugehörigen Sekundärindexblock als Erstes verwiesen wird.

SAM:

Der SETL positioniert den Block- und Satzzeiger auf die vom Anwender angegebene Stelle (Wiedergewinnungsadresse).

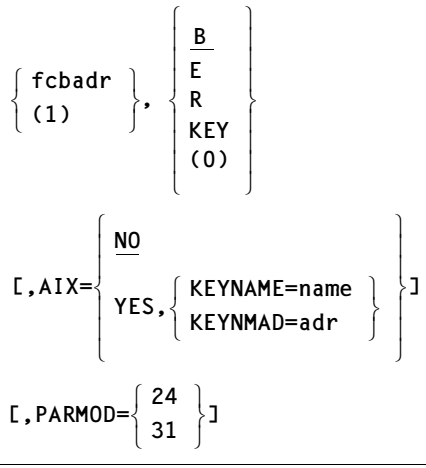
Der SETL-Makroaufruf positioniert den internen Satzzeiger. Der Anwender kann die Position definieren, an der die nachfolgende Bearbeitung der Datei beginnen soll.

Die Wiedergewinnungsadresse wird im 31-Bit-TU-FCB so verändert, dass sie nach einem folgenden GET- oder PUT-Makroaufruf korrekt versorgt ist; im 24-Bit-TU-FCB wird sie nicht verändert.

Da im 24-Bit-TU-FCB die Positionierungsinformation 1 Byte lang ist, dürfen in einem Puffer maximal 255 Sätze stehen.

Ein unzulässiger SETL-Operand bewirkt, dass die Steuerung an die Adresse USERERR des EXLST-Makroaufrufs abgegeben wird.

Format

Operation	Operanden
SETL	

Operandenbeschreibung

fcbadr
Adresse des FCB für die zu verarbeitende Datei.
Nur für ISAM:
Wenn in der Datei über einen Sekundärschlüssel positioniert werden soll, muss dieser FCB in seiner 31-Bit-Schnittstelle vorliegen.

(1)
Die FCB-Adresse steht im Register 1.

B
Es soll auf den Dateianfang positioniert werden.
Nur für ISAM:
Die Operation SETL B für eine Nulldatei führt auf den Fehlerausgang EOFADDR des EXLST-Makroaufrufs.
Nur für SAM:
Bei Mehrbanddateien wird auf den ersten Satz des aktuellen Bandes positioniert

E
Es soll auf das Dateiende positioniert werden.
Nur für SAM:
Bei Mehrbanddateien wird auf den letzten Satz des aktuellen Bandes positioniert, sodass der folgende GET-Makroaufruf Bandwechsel auslöst.
Bei OPEN OUTPUT/EXTEND führt die Angabe „E“ auf den Fehlerausgang USERERR des EXLST-Makros.

R
Nur für SAM:
Die Positionierungsinformation soll der Wiedergewinnungadresse entnommen werden (nicht zulässig für Banddateien mit Nichtstandardblöcken, die mit PARMOD=24 verarbeitet werden). Bei Mehrbanddateien bezieht sich die Wiedergewinnungsadresse auf das aktuelle Band, nicht auf die Datei.

KEY
Nur für ISAM:
Es soll auf den Primär- bzw. Sekundärschlüsselwert positioniert werden, der in dem über den KEYARG-Operanden des FCB bezeichneten Feld steht.

Verweist ein SETL ...,KEY auf einen existenten Schlüssel, lesen GET/GETR diesen Satz.
Verweist ein SETL ...,KEY auf einen nichtexistenten Schlüssel, liest ein folgender GET den Satz mit dem nächsthöheren Schlüssel, ein folgender GETR den Satz mit dem nächstniedrigeren Schlüssel.

(0)
Register 0 enthält einen „Positionierungs-Code“. Vor der Ausführung des SETL-Makroaufrufs muss das Register 0 versorgt werden.
Die folgende Tabelle zeigt den „Positionierungs-Code“ für Register 0.

SETL-Operand	Inhalt Register 0	Wirkung
B	0	Positionieren auf Dateianfang
E	1	Positionieren auf Dateiende
R	2	*nur bei SAM:* Position entsprechend Wiedergewinnungsadresse
KEY	Adresse von KEYARG	Positionieren auf bestimmten Satz

Enthält Register 0 einen anderen Wert als 0 oder 1, wird dieser immer als „KEYARG“-Adresse interpretiert.

AIX
Nur für ISAM:
Gibt an, ob auf einen Satz über seinen Primär- oder einen Sekundärschlüssel positioniert werden soll.

= NO
Auf den Satz wird über seinen Primärschlüssel positioniert. (Voreinstellung).

= YES
Kann nur angegeben werden, wenn
– die 31-Bit-Schnittstelle des Makros generiert wird (über den Operanden PARMOD=31 oder den Makroaufruf GPARMOD 31) und
– der Makro sich auf einen 31-Bit-FCB bezieht.

Auf den Satz wird über den im Operanden KEYNAME oder KEYNMAD vereinbarten Sekundärschlüssel positioniert.

KEYNAME = name
Nur für ISAM:
Gibt den Namen des Sekundärschlüssels an, über den auf einen Satz positioniert werden soll.
name muss der Name eines für die aktuelle Datei vereinbarten Sekundärschlüssels sein. Die Namen aller für eine Datei definierten Sekundärschlüssel können mit dem Makro SHOWAIX oder dem Kommando SHOW-INDEX-ATTRIBUTES ermittelt werden.

Es muss AIX=YES angegeben sein.

KEYNMAD = adr
Nur für ISAM:
Gibt die symbolische Adresse (den Namen) eines Feldes an, in dem der Anwender den Namen des Sekundärschlüssels hinterlegt hat, über den auf einen Satz positioniert werden soll.
Das Feld mit der symbolischen Adresse adr muss zum Zeitpunkt der Makroausführung den Namen eines für die aktuelle Datei vereinbarten Sekundärschlüssels enthalten.

Es muss AIX=YES angegeben sein.

PARMOD
Gibt den Generierungsmodus an.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24
Der Makroaufruf wird mit der Expansion für die 24-Bit-Schnittstelle aufgelöst. Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (24-Bit-Adressierungsmodus).

= 31
Der Makroaufruf wird adressierungsmodus-unabhängig (24-Bit- oder 31-Bit-Adressierung) generiert. Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist.

Hinweise zur Programmierung

1. Der SETL-Makroaufruf zerstört die Register 0, 1, 14 und 15.
2. Ein SETL-Makroaufruf führt immer zu einem SVC.

SHOPLNK – Informationen über ISAM-Pool-Kettungsnamen ausgeben

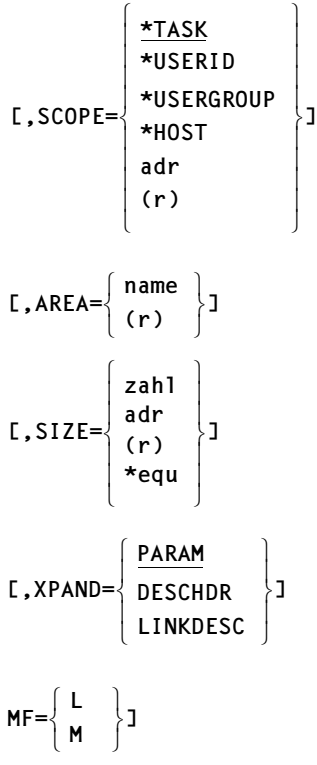
Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Der Makro SHOPLNK informiert über die Zuordnung von ISAM-Pools zu ISAM-Pool-Kettungsnamen. Hierzu kann der Anwender entweder den Pool-Kettungsnamen angeben oder den Namen des ISAM-Pools.
Ein ISAM-Pool-Kettungsname kann jedem ISAM-Pool zugeordnet werden, sofern ein Anschluss zu diesem ISAM-Pool besteht, unabhängig von Gültigkeitsbereich des betroffenden ISAM-Pools und vom Host-Rechner, an dem der betroffene ISAM-Pool vorhanden ist.

Format

Operation	Operanden
SHOPLNK	[,PARAM=adr] [,LINK={*ALL | 'name' | adr | (r)}] [,NAME={*ALL | 'name' | adr | (r)}] [,CATID={'name' | adr | (r)}]

(Teil 1 von 2)

Operation	Operanden
	
	MF=E,PARAM={ adr / (r) }
	MF=D[,PREFIX=pre]
	MF=C[,PREFIX=pre][,MACID=macid]

(Teil 2 von 2)

Operandenbeschreibung

AREA
Gibt einen Ausgabebereich an, in den die Ausgabeliste übertragen wird. Uneingeschränkter Zeichenvorrat und uneingeschränkte Länge.
Bei der Form MF=L kann nur der Operandenwert adr angegeben werden.

Voreinstellung: X'FFFFFFFF'

= adr
Symbolische Adresse eines Feldes, das den Namen des Ausgabereiches enthält.

= (r)
Register, das die Adresse eines Feldes enthält, in dem der Name des Ausgabebereiches hinterlegt ist.

CATID
Gibt die Katalogkennung des PVS an, dem der ISAM-Pool, zu dem Pool-Kettungsnamen ermittelt werden sollen, beim Kreieren zugeordnet wurde (nicht von Bedeutung bei Angabe des Parameters NAME=*ALL).
Bei der Form MF=L kann nur der Operandenwert 'name' angegeben werden.

Voreinstellung: die Default-PVS-Id des Auftrages (DEFCAT im JOIN-Eintrag) bzw. die Home-PVS-Id (abhängig von der Einstellung des Klasse-2-Systemparameters ISPLDEFC).

= '*name*'
Name eines Feldes mit der Katalogkennung des Pubsets. 4 Zeichen lang.

= adr
Symbolische Adresse eines Feldes mit der Katalogkennung des Pubsets.

= (r)
Register, das die Adresse eines Feldes mit der Katalogkennung des Pubsets enthält.

LINK = *ALL
Die Zuordnung aller vom Aufrufer definierten Pool-Kettungsnamen zu ISAM-Pools werden ermittelt.

= name
Gibt den ISAM-Pool-Kettungsnamen an, dessen Zuordnung zu einem ISAM-Pool ermittelt werden soll.

= adr
ist die symbolische Adresse eines Feldes, das den Link-Namen enthält.

= (r)
ist das Register, das die Adresse des LINK-Namens enthält.

MACID
wird nur in Verbindung mit MF=C ausgewertet und legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: MACID = ISL

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MF
die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

NAME
Gibt den Namen an, mit dem der ISAM-Pool, für den ISAM-Pool-Kettungsnamen ermittelt werden sollen, angelegt wurde. Der gewünschte ISAM-Pool wird eindeutig bestimmt über den angegebenen Namen, die Katalogkennung (CATID) und den Geltungsbereich (SCOPE).
Bei der Form MF=L können nur die Operandenwerte 'name'/*ALL angegeben werden.

= *ALL
Für alle vom Aufrufer erzeugten ISAM-Pools sollen die Pool-Kettungsnamen ermittelt werden.

= 'name'
Name des ISAM-Pools für den ISAM-Pool-Kettungsnamen ermittelt werden sollen.

= adr
Symbolische Adresse eines Feldes, das den Namen des ISAM-Pools oder *ALL enthält.

= (r)
Register, das die Adresse eines Feldes enthält, in dem der Name des ISAM-Pools oder *ALL hinterlegt ist.

SCOPE
Geltungsbereich des angegebenen ISAM-Pools, über den Informationen ausgegeben werden sollen. Wurde NAME=*ALL angegeben, werden Angaben beim Operanden SCOPE ignoriert.
Bei der Form MF=L können nur die Operandenwerte *TASK/*USERID/*HOST angegeben werden.

= *TASK
Für den tasklokalen ISAM-Pool mit dem angegebenen Namen sollen Pool-Kettungsnamen ausgegeben werden

= *USERID
= *USERGROUP
Die bis BS2000/OSD-BC V6.0A vorhandenen Geltungsbereiche SCOPE=USERID und SCOPE=USERGROUP werden aus Kompatibilitätsgründen noch akzeptiert, intern jedoch auf SCOPE=HOST (taskübergreifender ISAM-Pool) abgebildet.
Es werden aber jeweils nur die ISAM-Pools angezeigt, die mit dem angegebenen SCOPE angelegt wurden.

= *HOST
Für den taskübergreifenden ISAM-Pool mit dem angegebenen Namen sollen Pool-Kettungnamen ausgegeben werden

= adr
4 Byte lange symb. Adresse eines Feldes, das den Scope des ISAM-Pools enthält.

= (r)
Ist ein Register mit der Adresse, an der der Scope des ISAM-Pools hinterlegt ist.

SIZE
Gibt die Länge des Ausgabebereiches an.
Bei der Form MF=L können nur die Operandenwerte zahl/*equ angegeben werden.

Voreinstellung: X'00000000'

= zahl
numerische Zahl, die die Länge des Ausgabebereiches angibt: $100 \leq zahl \leq 10000$.

= adr
ist die 4 Byte lange symbolische Adresse für die Länge des Ausgabebereiches.

= (r)
Register, das die Länge des Ausgabebereiches enthält.

= *equ
EQUATE mit der Länge Ausgabereiches

XPAND
Steuerungs-Operand nur für MF=C und MF=D:
Es wird festgelegt, welche Struktur zu expandieren (erzeugen) ist. Angaben bei diesem Operanden werden bei anderen MF-Werten ignoriert.

= PARAM
Das Layout der Parameterliste wird expandiert.

= DESCHDR
Das Layout des Headers des Ausgabebereiches wird expandiert.

= LINKDESC
Das Layout eines Pool-Link-Deskriptors wird expandiert.

Returncodes

Die Returncodes werden im Header der jeweiligen Parameterliste hinterlegt:

– Der Main Code in einem Halbwort mit dem Namen DISLMRET.

– Der Subcode1 steht in einem Byte mit dem Namen DISLSR1. Er beschreibt Fehlerklassen, die es dem Aufrufer ermöglichen sollen, auf Fehlerklassen zu reagieren.

  Der Aufrufer kann sich sowohl am Maincode als auch am Subcode1 orientieren, wobei eine Auswertung des Subcode1 jedoch vorzuziehen ist: bei der Erweiterung der Maincodes für einen Makro sind Auswertungen nicht betroffen, wenn sie ausschließlich auf Fehlerklassen reagieren.

– Der Subcode2 hat immer den Wert X'00'.

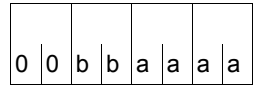
Falls Returncodes nicht im Header eines Makros abgelegt werden können, beispielsweise wenn dieser nicht zugreifbar ist, so wird das aufrufende Programm mit einer entsprechenden Fehlermeldung beendet.

Im Falle des Returncodes „interner Systemfehler beim Aufruf einer Systemfunktion“ enthält das Feld DISL.SYCD in der Parameterliste des jeweiligen Makros einen genaueren Code zur Diagnose (Werte: siehe Inserts bei der entsprechenden Message).

Ein Makro-Aufruf mit MF=D oder MF=C generiert zusätzlich zu den Feldnamen auch die EQU-Anweisungen für die Returncodes.

In der folgenden Übersicht werden für den Makro SHOPLNK die Returncodes in tabellarischer Form dargestellt. Die angegebenen Namen der EQU-Anweisungen sind links durch den String DISL zu ergänzen, wobei DISL modifiziert werden kann durch die Parameter PREFIX=prefix bzw. MACID=macid (siehe Operandenbeschreibung).

Standard-header:



Über die Ausführung des Makros SHOPLNK wird im Standardheader folgender Returncode übergeben (bb = SUBCODE1, aaaa = MAINCODE):

X'bb'	X'aaaa'	Erläuterung
X'00'	X'0000'	Funktion erfolgreich ausgeführt.
X'01'	X'FFFF'	Der Header ist fehlerhaft, z.B. nicht korrekt initialisiert. Keine Wiederholung möglich.
X'02'	X'FFFF'	Linkage Fehler (Function not available) Die aufgerufende Funktion steht nicht zur Verfügung (z.B. NK-ISAM nicht geladen)
X'03'	X'FFFF'	Linkage Fehler (Version not supported) Die im Header angegebene Version wird nicht unterstützt (Montage-Fehler)
X'01'	X'0001'	Parameterliste nicht zugreifbar. Keine Wiederholung möglich.
X'01'	X'0002'	Parameterfehler. Keine Wiederholung möglich.
X'20'	X'0005'	interner Systemfehler beim Aufruf einer Systemfunktion Keine Wiederholung möglich.
X'40'	X'0008'	Der angegebene ISAM-Pool-Link-Name existiert nicht. Fehler ungleich Fehlerklasse B, C, E.
X'40'	X'0009'	Der Aufrufer hat keinen ISAM-Pool-Link-Namen definiert Fehler ungleich Fehlerklasse B, C, E.
X'82'	X'000B'	kein virtueller Speicherplatz verfügbar. Warten und Wiederholen.

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Layout der Operandenliste

Makroauflösung mit MF=D und Standardwerten für PREFIX und MACID:

```
SHOPLNK MF=D
1          STACK  PRINT
1          PRINT  NOGEN
2               *,##### PREFIX=D, MACID=ISL #####
1          #INTF INTNAME=SHOPLNK,REFTYPE=REQUEST,INTCOMP=002
1 DISLPLA  DS    0F              BEGIN of PARAMETERAREA     _INOUT
1          FHDR  MF=(C,DISL),EQUATES=YES
2          DS    0A
2 DISLFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
2 *
2 DISLIFID DS    0A              0   INTERFACE IDENTIFIER
2 DISLFCTU DS    AL2             0   FUNCTION UNIT NUMBER
2 *                                  BIT 15    HEADER FLAG BIT,
2 *                                  MUST BE RESET UNTIL FURTHER NOTICE
2 *                                  BIT 14-12 UNUSED, MUST BE RESET
2 *                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
2 DISLFCT  DS    AL1             2   FUNCTION NUMBER
2 DISLFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
2 *
2 DISLRET  DS    0A              4   GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 DISLSRET DS    0AL2            4   SUB RETURN CODE
2 DISLSR2  DS    AL1             4   SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 DISLR2OK EQU   X'00'                All correct, no additional info
2 DISLR2NA EQU   X'01'                Successful, no action was necessary
2 DISLR2WA EQU   X'02'                Warning, particular situation
2 DISLSR1  DS    AL1             5   SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A    X'00'           FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B    X'01' - X'1F'   PARAMETER SYNTAX ERROR
2 * CLASS C    X'20'           INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D    X'40' - X'7F'   NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E    X'80' - X'82'   WAIT AND RETRY
2 *
2 DISLRFSP EQU   X'00'                FUNCTION SUCCESSFULLY PROCESSED
2 DISLRPER EQU   X'01'                PARAMETER SYNTAX ERROR
2 *  3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
```

```
2 DISLRFNS EQU   X'01'               CALLED FUNCTION NOT SUPPORTED
2 DISLRFNA EQU   X'02'               CALLED FUNCTION NOT AVAILABLE
2 DISLRVNA EQU   X'03'               INTERFACE VERSION NOT SUPPORTED
2 *
2 DISLRAER EQU   X'04'               ALIGNMENT ERROR
2 DISLRIER EQU   X'20'               INTERNAL ERROR
2 DISLRCAR EQU   X'40'               CORRECT AND RETRY
2 *  2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 DISLRECR EQU   X'41'               SUBSYSTEM (SS) MUST BE CREATED
2 *                                  EXPLICITELY BY CREATE-SS
2 DISLRECN EQU   X'42'               SS MUST BE EXPLICITELY CONNECTED
2 *
2 DISLRWAR EQU   X'80'               WAIT FOR A SHORT TIME AND RETRY
2 DISLRWLR EQU   X'81'                    "     LONG       "
2 DISLRWUR EQU   X'82'               WAIT TIME IS UNCALCULABLY LONG
2 *                                  BUT RETRY IS POSSIBLE
2 *  2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 DISLRTNA EQU   X'81'               SS TEMPORARILY NOT AVAILABLE
2 DISLRDH  EQU   X'82'               SS IN DELETE / HOLD
2 *
2 DISLMRET DS    0AL2            6   MAIN RETURN CODE
2 DISLMR2  DS    AL1             6   MAIN RETURN CODE 2
2 DISLMR1  DS    AL1             7   MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XXYYYY')
2 *
2 DISLRLNK EQU   X'FFFF'              LINKAGE ERROR / REQ. NOT PROCESSED
2 DISLFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
2 *
1 DISLOK   EQU   X'0000'            FUNCTION SUCCESSFUL PROCESSED
1 DISLNPAR EQU   X'0001'            PARAMETERLIST NOT ACCESSIBLE
1 DISLPERR EQU   X'0002'            PARAMETER ERROR
1 DISLSYSE EQU   X'0005'            INTERNAL SYSTEM ERROR
1 DISLLLNE EQU   X'0008'            SPECIFIED POOL LINK NAME NOT FOUND
1 DISLNOLI EQU   X'0009'            NO ISAM POOL LINK EXISTING
1 DISLNOSP EQU   X'000B'            NO MEMORY AVAILABLE
1 DISLNUGR EQU   X'000C'            NO USEGROUP DEFINED
1 DISLINOP EQU   X'001F'            SSTA INOP
1 DISLSSER EQU   X'0020'            SSTA INTERNAL ERROR
1 DISLMEMR EQU   X'0021'            SSTA MEMORY ERROR
1 DISLOPSR EQU   X'0022'            SSTA OPS ERROR
1 DISLOPME EQU   X'0023'            SSTA OPS MEMORY ERROR
1 *
1 *
1 DISLPLNK DS    CL8                LINK-NAME
1 DISLPNAM DS    CL8                POOL-NAME
1 DISLCID  DS    CL4                CATALOG-IDENTIFIER
1 *
```

```
1 DISLSCOP DS    XL1              SCOPE
1 DISLTASK EQU   X'00'                 = TASK
1 DISLUSID EQU   X'01'                 = USERID
1 DISLHOST EQU   X'02'                 = HOST-SYSTEM
1 DISLUSGR EQU   X'03'                 = USERGROUP
1 *
1 DISLSYCD DS    XL1              SYSTEM-ERROR-CODE
1 *
1 DISLADDR DS    A                ADDRESS OF OUTPUT-AREA
1 DISLSIZE DS    F                SIZE OF OUTPUT-AREA
1 *
1 DISL#    EQU   *-DISLPLA        LENGTH of PARAMETERAREA
```

Format des Ausgabebereiches

Der Makro SHOPLNK liefert dem Aufrufer die Zuordnungen von ISAM-Pools zu ISAM-Pool-Kettungsnamen in seinem Ausgabe-Bereich. Dieser beginnt zunächst mit einem Verwaltungs-Header der Länge 16 Byte.

Der Verwaltungs-Header des Ausgabe-Bereiches wird generiert durch die Angabe des Steuer-Operanden XPAND=DESCHDR beim Aufruf des Makros SHOPLNK mit dem Steuer-Operanden MF=D oder MF=C:

```
1 DISLADMH DS    0F               VERWALTUNGS-HEADER-LAYOUT
1 DISLLLG  DS    F                ANZAHL UEBERTRAGENER BYTES
1 DISLLCLG DS    F                LAENGE DER GESAMT-INFORMATION
1 DISLL#LN DS    H                ANZAHL BETROFFENER LINKNAMEN
1 DISLLIND DS    XL1              UEBERTRAGUNGS-INDIKATOR
1 DISLLCOM EQU   X'00'            INFORMATION VOLLSTAENDIG
1 DISLLPAR EQU   X'01'            INFORMATION UNVOLLSTAENDIG
1 DISLLRES DS    CL5              NICHT VERWENDET
1 DISLLLEN EQU   *-DISLADMH       LAENGE DES HEADERS
```

Ein Eintrag mit Informationen wird expandiert durch die Angabe des Parameters XPAND=LINKDESC beim Aufruf des Makros SHOPLNK mit dem Steuer-Operanden MF=C oder MF=D:

```
1 DISLLDDS DS    0F               POOL-LINK-DESCRIPTOR-LAYOUT
1 DISLLNAM DS    CL8              NAME DES POOL-LINKS
1 DISLPONA DS    CL8              NAME DES ZUGEORDNETEN ISAM-POOLS
1 DISLLCID DS    CL4              NAME DES ZUGEORDNETEN PVS
1 DISLLSCO DS    XL1              SCOPE DES ISAM-POOLS
1 DISLLTSK EQU   X'00'            SCOPE = TASK
1 DISLLUSR EQU   X'01'            SCOPE = USERID
1 DISLLHOS EQU   X'02'            SCOPE = HOST
1 DISLLUGR EQU   X'03'            SCOPE = USERGROUP
1 DISLLUID DS    CL8              USERID BEI SCOPE = *USERID
1 *                               GRUPPENNAME BEI SCOPE = *USERGROUP
```

```
1 DISLLRSV DS    CL3              NICHT VERWENDET
1 DISLLLNG EQU   *-DISLLDDS       LAENGE DES ISAM-POOL-DESCR.
```

Erläuterungen zu den einzelnen Feldern des Ausgabebereiches

Verwaltungsheader

Der Verwaltungs-Header (Länge 16 Byte) enthält folgende Informationen:

– die Anzahl der Bytes, die in den Ausgabebereich übertragen wurde (Länge: 4 Byte),

– die Anzahl der Bytes, aus denen die gesamte Information besteht (Länge: 4 Byte; dies ist dann für den Aufrufer des Makros von Bedeutung, wenn der Ausgabebereich zu klein ist: in diesem Falle muss der Aufrufer des Makros den Ausgabebereich in dieser Länge (mindestens) zur Verfügung stellen),

– die Anzahl der ISAM-Pool-Kettungsnamen,für die im Ausgabebereich die Zuordnung zu ISAM-Pools ermittelt wurde (Länge: 2 Byte),

– ein Indikator der Länge 1 Byte, der angibt, ob die in den Ausgabebereich übertragene Information vollständig ist oder ob auf Grund eines zu kurzen Ausgabebereiches nur ein Teil der gewünschten Informationen geliefert werden konnte, wobei gilt:

  X'00' -- Information ist komplett,

  X'01' -- Information wurde abgeschnitten und ist unvollständig;

  (im letzteren Falle muss der Aufrufer einen größeren Ausgabebereich zur Verfügung stellen, um die komplette Information zu erhalten; die Länge des größeren Ausgabebereiches ist dann dem zweiten Wort des Verwaltungs-Headers zu entnehmen),

– die restlichen 5 Byte werden nicht verwendet.

Deskriptorbereich

Zur Identifikation eines ISAM-Pools in eindeutiger Weise benötigt man den Namen des ISAM-Pool sowie die CATID des Host-Rechners, an dem der ISAM-Pool vorhanden ist, und außerdem den Gültigkeitsbereich des ISAM-Pools. Der Ausgabebereich für den Makro SHOPLNK enthält Einträge in der Länge von 32 Byte, die zunächst einen ISAM-Pool-Kettungsnamen enthalten, gefolgt von einem ISAM-Pool-Descriptor mit den oben genanntan Informationen. Somit sind in einem Eintrag im Ausgabebereich folgende Informationen enthalten:

– der Name eines ISAM-Pool-Kettungsnamens in der Länge von 8 Byte (abdruckbar),

– der Name des zugeordneten ISAM-Pools in der Länge von 8 Byte (abdruckbar),

– die CATID des Host-Rechners, auf dem der betroffene ISAM-Pool vorhanden ist, ebenfalls abdruckbar in der Länge von 4 Byte,

– den SCOPE (Gültigkeits-Bereich) des betroffenen ISAM-Pools in der Länge 1 Byte, wobei gilt:

  X'00' -- SCOPE = *TASK

  X'01' -- SCOPE = *USERID

  X'02' -- SCOPE = *HOST

  X'03' -- SCOPE = *USERGROUP

– die USERID, falls der zugeordnete ISAM-Pool mit dem Attribut SCOPE = USERID versehen ist,

– die restlichen 3 Byte werden nicht verwendet.

Der Ausgabebereich, den der Makro SHOPLNK versorgt, hat mit den genannten Strukturen dann den folgenden Aufbau:

HEADER					
LINKNAME1	POOLNAME1	CATID1	SCOPE1	userid	UNUSED
LINKNAME2	POOLNAME2	CATID2	SCOPE2	userid	UNUSED
. . . . .					
LINKNAMEn	POOLNAMEn	CATIDn	SCOPEn	userid	UNUSED

Das Feld „userid“ enthält Blanks bei SCOPE=TASK oder SCOPE=HOST.

Im Anschluss an den Verwaltungs-Header wird die Zuordnung von ISAM-Pools zu Pool-Kettungsnamen beschrieben; die Anzahl der erstellten Einträge ist im Verwaltungs-Header enthalten.

SHOPOOL – Informationen über ISAM-Pool ausgeben

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form/M-Form) (siehe Seite 870)

Der Makro SHOPOOL gibt Informationen über ISAM-Pools aus, mit denen der Auftrag gerade verbunden ist. Verbindungen zu ISAM-Pools auf Remote-Rechnern werden berücksichtigt (Ausnahme beim Operanden SELECT).
Der Benutzer kann die Informationen sowohl für einen bestimmten ISAM-Pool, als auch für alle ISAM-Pools anfordern, zu dem bzw. zu denen eine Verbindung existiert. Mit SHOPOOL werden alle Pool-spezifischen Eigenschaften je ISAM-Pool (wie im Makro CREPOOL vereinbart) ausgegeben.
Zusätzlich kann sich der Benutzer für jeden Pool die Auftragsnummern (TSNs) der angeschlossenen Aufträge ausgeben lassen.
Den Anschluss an einen ISAM-Pool veranlasst der Benutzer mit dem Makro CREPOOL. Der Auftrag kann auch implizit von NK-ISAM an Standard-Pools angeschlossen sein.

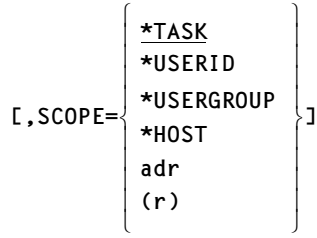
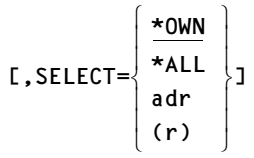
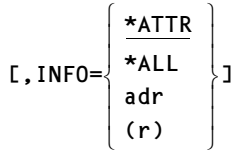
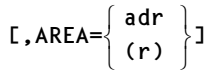
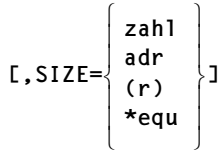
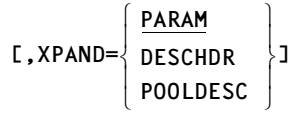
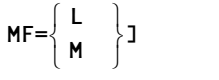
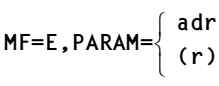
Hinweis

Taskübergreifende ISAM-Pools (SCOPE=HOST) werden bei der Dateieröffnung automatisch dateispezifisch in einem Data Space angelegt.
Die bis BS2000/OSD-BC V6.0A vorhandenen Geltungsbereiche SCOPE=USERID und SCOPE=USERGROUP werden aus Kompatibilitätsgründen noch akzeptiert, intern jedoch auf SCOPE=HOST (taskübergreifender ISAM-Pool) abgebildet. Für weitere Informationen zu ISAM-Pools in Data Spaces siehe Handbuch „Einführung in das DVS“ [1].

Format

Operation	Operanden
SHOPOOL	[,PARAM=adr] [,NAME={'name' | *ALL | adr | (r)}] [,CATID={'name' | adr | (r)}]

(Teil 1 von 2)

Operation	Operanden
	
	
	
	
	
	
	
	
	MF=D[,PREFIX=pre]
	MF=C[,PREFIX=pre][,MACID=macid]

(Teil 2 von 2)

Operandenbeschreibung

AREA
Gibt einen Ausgabebereich an, in den die Ausgabeliste übertragen wird.
Bei der Form MF=L kann nur der Operandenwert adr angegeben werden.

Voreinstellung: X'FFFFFFFF'

= adr
Symbolische Adresse (Name) des Ausgabebereiches. Uneingeschränkter Zeichenvorrat und uneingeschränkte Länge.

= (r)
Register, in das die Adresse des Ausgabebereiches eingetragen wird.

CATID
Katalogkennung des Pubsets, dem der ISAM-Pool, über den Informationen ausgegeben werden sollen, zugeordnet ist (nicht von Bedeutung bei Angabe des Parameters NAME= *ALL). Bei der Form MF=L kann nur der Operandenwert 'name' angegeben werden.

Voreinstellung: Der ISAM-Pool ist dem Katalog zugeordnet, der mit dem Klasse-2-Systemparameter ISPLDEFC (ISAM-POOL-DEFAULT-CATID) bei Systemgenerierung eingestellt wurde: ,

X'00':	Standard-Katalogkennung aus dem Benutzereintrag (DEFCAT)
X'01':	Katalogkennung des Home-Pubsets

= 'name'
Katalogkennung des Pubsets. 4 Zeichen lang.

= adr
4 Byte lange Adresse eines Feldes, das die Katalogkennung des Pubsets enthält.

= (r)
ist das Register das die Adresse eines Feldes mit der Katalogkennung des Pubsets.

INFO
Bestimmt den Umfang der auszugebenden Information. Der Makro SHOPOOL liefert die gewünschten Informationen in den vom Aufrufer angegebenen Ausgabebereich (siehe „Format des Ausgabebereiches“ auf Seite 841).

= *ATTR
Für den im Operanden NAME angegebenen ISAM-Pool werden die „statischen“ Eigenschaften ausgegeben.

= *ALL
Zusätzlich zu den statischen Eigenschaften werden die Auftragsnummern (TSN's) aller Aufträge ausgegeben, die an den/die angegebenen ISAM-Pool/s angeschlossen sind.

= adr
Ist die Adresse eines Feldes, in dem der Umfang der auszugebenen Informationen hinterlegt ist.

= (r)
Ist das Register, in dem die Adresse eines Feldes mit dem Umfang der auszugebenden Informationen abgelegt ist.

MACID
Wird nur in Verbindung mit MF=C ausgewertet und legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: MACID = ISP

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

MF
die Formen des MF-Operanden sind detailliert im Anhang (Seite 870) beschrieben.

NAME
Name des ISAM-Pools, über den informiert werden soll. Der gewünschte ISAM-Pool wird eindeuting bestimmt über den angegebenen Namen, die Katalogkennung (CATID) und den Geltungsbereich (SCOPE).
Informationen werden nur ausgegeben, wenn der ISAM-Pool existiert und der aktuelle Auftrag an ihn angeschlossen ist.
Bei der Form MF=L können nur die Operandenwerte 'name'/*ALL angegeben werden.

= *ALL
Informiert über alle ISAM-Pools, an die der aktuelle Auftrag angeschlossen ist.

= 'name'
Name des ISAM-Pools, über den informiert werden soll. 'name' kann 8 Zeichen lang sein.

= adr
Adresse eines Feldes, das den Namen des ISAM-Pools oder *ALL enthält.

= (r)
Register, das eine Adresse enthält, an welcher der Name des ISAM-Pools oder *ALL hinterlegt ist.

PARAM
bezeichnet die Adresse der Operandenliste und wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

= adr
adr ist die symbolische Adresse (Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

PREFIX
wird nur in Verbindung mit MF=C oder MF=D ausgewertet und legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung im Datenbereich generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein Zeichen langes Präfix, mit dem die vom Assembler generierten Feldnamen und Equates beginnen sollen.

SCOPE
Geltungsbereich des angegebenen ISAM-Pools, über den Informationen ausgegeben werden sollen. Wurde NAME=*ALL angegeben, werden Angaben beim Operanden SCOPE ignoriert.
Bei der Form MF=L können nur die Operandenwerte *TASK/*USER-ID/*HOST angegeben werden.

= *TASK
Informiert über den Task-spezifischen ISAM-Pool mit dem angegebenen Namen.

= *USERID
= *USERGROUP
Die bis BS2000/OSD-BC V6.0A vorhandenen Geltungsbereiche SCOPE=USERID und SCOPE=USERGROUP werden aus Kompatibilitätsgründen noch akzeptiert, intern jedoch auf SCOPE=HOST (taskübergreifender ISAM-Pool) abgebildet.
Es werden aber jeweils nur die ISAM-Pools angezeigt, die mit dem angegebenen SCOPE angelegt wurden.

= *HOST
Informiert über den taskübergreifenden ISAM-Pool mit dem angegebenen Namen.

= adr
Symbolischer Name eines Feldes, das den Geltungsbereich des angegebenen ISAM-Pools enthält.

= (r)
Register mit der Adresse eines Feldes, das den Geltungsbereich des ISAM-Pools enthält.

SELECT
Gibt an, nach welchen Kriterien die durch den Parameter NAME spezifizierten ISAM-Pools zur Ausgabe von Informationen ausgewählt werden.

= *OWN
Es werden Informationen über ISAM-Pools ausgegeben, zu denen die aufrufende Task einen Anschluss besitzt. Die Ausgabe umfasst dabei auch ISAM-Pools auf Remote-Rechnern.

= *ALL
Dieser Operandenwert kann nur von einer Task spezifiziert werden, die über das Privileg TSOS oder über das Privileg SW-MONITOR-ADMINISTRATION verfügt.
Es werden Informationen über alle vorhandenen ISAM-Pools ausgegeben, selbst wenn kein Anschluss zu diesen ISAM-Pools vorhanden ist. Remote ISAM-Pools werden nicht erfasst.

= adr
Symbolischer Name eines Feldes, das Informationen über die Kriterien enthält nach denen Informationen über ISAM-Pools ausgegeben werden.

= (r)
Register mit der Adresse eines Feldes, das Informationen über die Kriterien enthält nach denen Informationen über ISAM-Pools ausgegeben werden.

SIZE
Gibt die Länge des Ausgabebereiches an.

Bei der Form MF=L können nur die Operandenwerte zahl/*equ angegeben werden.

Voreinstellung: X'00000000'

= zahl
Ist eine numerische Zahl, die die Länge des Ausgabebereiches angibt.
$100 \leq$ zahl ≤ 10000.

= adr
Ist die Adresse für die Länge des Ausgabebereiches.

= *equ
EQUATE, das die Länge des Ausgabebereiches beinhaltet.

XPAND

Steuerungs-Operand nur für MF=C und MF=D:

Es wird festgelegt, welche Struktur zu expandieren (erzeugen) ist. Angaben bei diesem Operanden werden bei anderen MF-Werten ignoriert.

= PARAM

Das Layout der Parameterliste wird expandiert.

= DESCHDR

Das Layout des Headers des Ausgabebereiches wird expandiert.

= POOLDESC

Das Layout eines Pool-Deskriptors wird expandiert.

Returncodes

Die Returncodes werden im Header der jeweiligen Parameterliste hinterlegt:

– Der Main Code in einem Halbwort mit dem Namen DISPMRET.

– Der Subcode1 steht in einem Byte mit dem Namen DISPSR1. Er beschreibt Fehlerklassen, die es dem Aufrufer ermöglichen sollen, auf Fehlerklassen zu reagieren.

  Der Aufrufer kann sich sowohl am Maincode als auch am Subcode1 orientieren, wobei eine Auswertung des Subcode1 jedoch vorzuziehen ist: bei der Erweiterung der Maincodes für einen Makro sind Auswertungen nicht betroffen, wenn sie ausschließlich auf Fehlerklassen reagieren.

– Der Subcode2 hat immer den Wert X'00'.

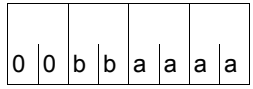
Falls Returncodes nicht im Header eines Makros abgelegt werden können, beispielsweise wenn dieser nicht zugreifbar ist, so wird das aufrufende Programm mit einer entsprechenden Fehlermeldung beendet.

Im Falle des Returncodes „interner Systemfehler beim Aufruf einer Systemfunktion“ enthält das Feld DISPSYCD in der Parameterliste des jeweiligen Makros einen genaueren Code zur Diagnose (Werte: siehe Inserts bei der entsprechenden Meldung).

Ein Makro-Aufruf mit MF=D oder MF=C generiert zusätzlich zu den Feldnamen auch die EQU-Anweisungen für die Returncodes.

In der folgenden Übersicht werden für den Makro SHOPOOL die Returncodes in tabellarischer Form dargestellt. Die angegebenen Namen der EQU-Anweisungen sind links durch den String DISP zu ergänzen, wobei DISP modifiziert werden kann durch die Parameter PREFIX=prefix bzw. MACID=macid (siehe Operandenbeschreibung).

Standard-header:



Über die Ausführung des Makros SHOPOOL wird im Standardheader folgender Returncode übergeben (bb = SUBCODE1, aaaa = MAINCODE):

X'bb'	**X'aaaa'**	**Erläuterung**
X'00'	X'0000'	Funktion erfolgreich ausgeführt.
X'01'	X'FFFF'	Der Header ist fehlerhaft, z.B. nicht korrekt initialisiert. Keine Wiederholung möglich.
X'02'	X'FFFF'	Linkage Fehler (Function not available) Die aufgerufende Funktion steht nicht zur Verfügung
X'03'	X'FFFF'	Linkage Fehler (Version not supported) Die im Header angegebene Version wird nicht unterstützt (Montage-Fehler)
X'01'	X'0001'	Parameterliste nicht zugreifbar. Keine Wiederholung möglich.
X'01'	X'0002'	Parameterfehler. Keine Wiederholung möglich.
X'40'	X'0003'	spezifizierte CATID existiert nicht. Fehlerklasse ungleich B, C, E.
X'40'	X'0004'	Der spezifizierte ISAM-Pool existiert nicht Fehlerklasse ungleich B, C, E.
X'20'	X'0005'	interner Systemfehler beim Aufruf einer Systemfunktion Keine Wiederholung möglich.
X'40'	X'0006'	kein ISAM-Pool vorhanden Fehlerklasse ungleich B, C, E.
X'40'	X'0007'	Funktion ist nicht mit dem nötigen Privileg ausgestattet. Fehlerklasse ungleich B, C, E.
X'82'	X'000A'	spezifizierte CATID nicht verfügbar. Warten und Wiederholen.
X'82'	X'000B'	kein virtueller Speicherplatz verfügbar. Warten und Wiederholen.

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Aufbau der Parameterliste

Makroauflösung mit MF=D und Standardwerten für PREFIX und MACID:

Die Parameterliste des Makros enthält einen Header, dessen Felder beim Aufbau der Parameterliste mithilfe der L-Form automatisch versorgt werden.

Wird eine Parameterliste mit der D-Form oder C-Form dynamisch aufgebaut, so ist sie zuvor durch eine mithilfe der L-Form erzeugten Parameterliste zu initialisieren; nur auf diese Weise ist eine korrekte Versorgung des Headers einer Parameterliste gewährleistet.

```
SHOPOOL MF=D
1          STACK  PRINT
1          PRINT  NOGEN
2               *,##### PREFIX=D, MACID=ISP #####
1          #INTF INTNAME=SHOPOOL,REFTYPE=REQUEST,INTCOMP=002
1 DISPPPA  DS    0F                BEGIN of PARAMETERAREA     _INOUT
1          FHDR  MF=(C,DISP),EQUATES=YES
2          DS    0A
2 DISPFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
2 *
2 DISPIFID DS    0A              0   INTERFACE IDENTIFIER
2 DISPFCTU DS    AL2             0   FUNCTION UNIT NUMBER
2 *                                  BIT 15    HEADER FLAG BIT,
2 *                                  MUST BE RESET UNTIL FURTHER NOTICE
2 *                                  BIT 14-12 UNUSED, MUST BE RESET
2 *                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
2 DISPFCT  DS    AL1             2   FUNCTION NUMBER
2 DISPFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
2 *
2 DISPRET  DS    0A              4   GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 DISPSRET DS    0AL2            4   SUB RETURN CODE
2 DISPSR2  DS    AL1             4   SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 DISPR2OK EQU   X'00'               All correct, no additional info
2 DISPR2NA EQU   X'01'               Successful, no action was necessary
2 DISPR2WA EQU   X'02'               Warning, particular situation
2 DISPSR1  DS    AL1             5   SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A    X'00'           FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B    X'01' - X'1F'   PARAMETER SYNTAX ERROR
2 * CLASS C    X'20'           INTERNAL ERROR IN CALLED FUNCTION
```

```
2 * CLASS D    X'40' - X'7F'   NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E    X'80' - X'82'   WAIT AND RETRY
2 *
2 DISPRFSP EQU   X'00'              FUNCTION SUCCESSFULLY PROCESSED
2 DISPRPER EQU   X'01'              PARAMETER SYNTAX ERROR
2 *  3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 DISPRFNS EQU   X'01'              CALLED FUNCTION NOT SUPPORTED
2 DISPRFNA EQU   X'02'              CALLED FUNCTION NOT AVAILABLE
2 DISPRVNA EQU   X'03'              INTERFACE VERSION NOT SUPPORTED
2 *
2 DISPRAER EQU   X'04'              ALIGNMENT ERROR
2 DISPRIER EQU   X'20'              INTERNAL ERROR
2 DISPRCAR EQU   X'40'              CORRECT AND RETRY
2 *  2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 DISPRECR EQU   X'41'              SUBSYSTEM (SS) MUST BE CREATED
2 *                                 EXPLICITELY BY CREATE-SS
2 DISPRECN EQU   X'42'              SS MUST BE EXPLICITELY CONNECTED
2 *
2 DISPRWAR EQU   X'80'              WAIT FOR A SHORT TIME AND RETRY
2 DISPRWLR EQU   X'81'                   "     LONG       "
2 DISPRWUR EQU   X'82'              WAIT TIME IS UNCALCULABLY LONG
2 *                                 BUT RETRY IS POSSIBLE
2 *  2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 DISPRTNA EQU   X'81'              SS TEMPORARILY NOT AVAILABLE
2 DISPRDH  EQU   X'82'              SS IN DELETE / HOLD
2 *
2 DISPMRET DS    0AL2           6   MAIN RETURN CODE
2 DISPMR2  DS    AL1            6   MAIN RETURN CODE 2
2 DISPMR1  DS    AL1            7   MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XXYYYY')
2 *
2 DISPRLNK EQU   X'FFFF'             LINKAGE ERROR / REQ. NOT PROCESSED
2 DISPFHL  EQU   8              8   GENERAL OPERAND LIST HEADER LENGTH
2 *
1 DISPOK   EQU   X'0000'           FUNCTION SUCCESSFUL PROCESSED
1 DISPNPAR EQU   X'0001'           PARAMETERLIST NOT ACCESSIBLE
1 DISPPERR EQU   X'0002'           PARAMETER ERROR
1 DISPNCAT EQU   X'0003'           CATID NOT KNOWN
1 DISPPLNE EQU   X'0004'           SPECIFIED ISAM POOL NOT FOUND
1 DISPSYSE EQU   X'0005'           INTERNAL SYSTEM ERROR
1 DISPNOPL EQU   X'0006'           NO ISAM POOL EXISTING
1 DISPNAUT EQU   X'0007'           NO AUTHORIZATION FOR FUNCTION
1 DISPNACC EQU   X'000A'           CATID NOT AVAILABLE
1 DISPNOSP EQU   X'000B'           NO MEMORY AVAILABLE
1 DISPNUGR EQU   X'000C'           NO USERGROUP DEFINED
1 DISPINOP EQU   X'001F'           SSTA INOP
1 DISPSSER EQU   X'0020'           SSTA INTERNAL ERROR
```

```
1 DISPMEMR EQU   X'0021'          SSTA MEMORY ERROR
1 DISPOPSR EQU   X'0022'          SSTA OPS ERROR
1 DISPOPME EQU   X'0023'          SSTA OPS MEMORY ERROR
1 *
1 *
1 DISPPNAM DS    CL8              POOL-NAME
1 DISPCID  DS    CL4              CATALOG-IDENTIFIER
1 *
1 DISPSCOP DS    XL1              SCOPE
1 DISPTASK EQU   X'00'                  = TASK
1 DISPUSID EQU   X'01'                  = USERID
1 DISPHOST EQU   X'02'                  = HOST-SYSTEM
1 DISPUSGR EQU   X'03'                  = USERGROUP
1 *
1 DISPSELC DS    XL1              SELECT
1 DISPOWN  EQU   X'00'                  = OWN
1 DISPALLH EQU   X'01'                  = ALL (HOST)
1 *
1 DISPINFO DS    XL1              INFO
1 DISPATTR EQU   X'00'                  = ATTR
1 DISPALLT EQU   X'01'                  = ALL
1 *
1 DISPSYCD DS    XL1              SYS-ERROR-CODE
1 *
1 DISPADDR DS    A                ADDRESS OF OUTPUT-AREA
1 DISPSIZE DS    F                SIZE OF OUTPUT-AREA
1 *
1 DISP#    EQU   *-DISPPPA        LENGTH of PARAMETERAREA
```

Format des Ausgabebereiches

Der Makro SHOPOOL liefert dem Aufrufer die gewünschten Informationen in dessen spezifiziertem Ausgabebereich.

Verwaltungsheader des Ausgabebereiches

Der Ausgabebereich beginnt mit einem Verwaltungs-Header der Länge 16 Byte. Der Verwaltungs-Header des Ausgabebereiches wird generiert durch Angabe des Operanden XPAND=DESCHDR beim Aufruf des Makros SHOPOOL mit dem Steuerparameter MF=C oder MF=D:

```
1 DISPADMH DS    0F              VERWALTUNGS-HEADER-LAYOUT
1 DISPPLG  DS    F               ANZAHL UEBERTRAGENER BYTES
1 DISPPCLG DS    F               LAENGE DER GESAMT-INFORMATION
1 DISPP#PO DS    H               ANZAHL BETROFFENER ISAM-POOLS
1 DISPPINF DS    XL1             INDIKATOR FUER INFO-UMFANG
1 DISPPATT EQU   X'00'           INFO = *ATTR
1 DISPPALL EQU   X'01'           INFO = *ALL
1 DISPPIND DS    XL1             UEBERTRAGUNGS-INDIKATOR
1 DISPPCOM EQU   X'00'           INFORMATION VOLLSTAENDIG
1 DISPPPAR EQU   X'01'           INFORMATION UNVOLLSTAENDIG
1 DISPPRES DS    CL4             NICHT VERWENDET
1 DISPPLEN EQU   *-DISPADMH      LAENGE DES HEADERS
```

Spezifizierter Ausgabebereich

Im spezifierten Ausgabebereichs des Aufrufers wird ein ISAM-Pool beschrieben durch eine sog. ISAM-Pool-Deskriptor in der Länge von 32 Byte. Ein ISAM-Pool-Deskriptor wird generiert durch die Angabe des Steueroperanden XPAND=POOLDESC beim Aufruf des Makros SHOPOOL mit dem Steueroperand MF=C oder MF=D:

```
1 DISPPDDS DS    0F              ISAM-POOL-DESCRIPTOR-LAYOUT
1 DISPNAME DS    CL8             NAME DES ISAM-POOLS
1 DISPPCID DS    CL4             NAME DES ZUGEORDNETEN PVS
1 DISPPSIZ DS    F               POOL-GROESSE (IN 2K-EINHEITEN)
1 DISPPSCO DS    XL1             SCOPE DES ISAM-POOLS
1 DISPPTSK EQU   X'00'           SCOPE = TASK
1 DISPPUSR EQU   X'01'           SCOPE = USERID
1 DISPPHOS EQU   X'02'           SCOPE = HOST
1 DISPPUGR EQU   X'03'           SCOPE = USERGROUP
1 DISPPWRO DS    XL1             WROUT-EIGENSCHAFT DES ISAM-POOLS
1 DISPPDEF EQU   X'00'           WROUT = DEFERRED
1 DISPPIMM EQU   X'01'           WROUT = IMMEDIATE
1 DISPPCST DS    XL1             CSTAT-RESIDENZ DES ISAM-POOLS
1 DISPPNRE EQU   X'00'           ISAM-POOL NICHT CSTAT-RESIDENT
1 DISPPRSD EQU   X'01'           ISAM-POOL CSTAT-RESIDENT
1 DISPPEXT DS    XL1             VORHANDENE ISAM-POOL-EXTENTS
```

```
1 DISPPNEX EQU   X'00'            KEIN EXTENT VORHANDEN
1 DISPPEX2 EQU   X'01'            2K-EXTENT VORHANDEN
1 DISPPEX4 EQU   X'02'            4K-EXTENT VORHANDEN
1 DISPPEXA EQU   X'03'            2K- UND 4K-EXTENT VORHANDEN
1 DISPPLCI DS    XL1              LOKALITAETS-INDIKATOR
1 DISPPLCL EQU   X'00'            POOL AUF LOKALEM RECHNER
1 DISPPREM EQU   X'01'            POOL AUF REMOTE RECHNER
1 DISPPUID DS    CL8              USERID BEI SCOPE = *USERID
1 *                               GRUPPENNAME BEI SCOPE = *USERGROUP
1 DISPPRSV DS    CL3              NICHT VERWENDET
1 DISPPLNG EQU   *-DISPPDDS       LAENGE DES ISAM-POOL-DESCRIPTORS
```

Erläuterungen zum Aufbau des Ausgabebereichs

Der **Verwaltungs-Header** enthält die folgenden Informationen:

– die Anzahl der Bytes, die in den Ausgabebereich übertragen wurde (Länge: 4 Byte),

– die Anzahl der Bytes, aus denen die gesamte geforderte Information besteht (Länge: 4 Byte; der Aufrufer des Makros muss den Ausgabebereich in dieser Länge zur Verfügung stellen),

– die Anzahl der ISAM-Pools, über die im Ausgabebereich Informationen hinterlegt wurden (Länge: 2 Byte),

– ein Indikator der Länge 1 Byte, der angibt, welcher Informationsumfang beim Makro-Aufruf spezifiziert wurde, wobei gilt:

  X'00' -- INFO = *ATTR

  X'01' -- INFO = *ALL

– ein Indikator der Länge 1 Byte, der angibt, ob die in den Ausgabebereich übertragene Information vollständig ist oder ob auf Grund eines zu kurzen Ausgabebereiches nur ein Teil der gewünschten Informationen geliefert werden konnte, wobei gilt:

  X'00' -- Information ist komplett,

  X'01' -- Information wurde abgeschnitten und ist unvollständig;

  (im letzteren Falle muss der Aufrufer einen größeren Ausgabebereich zur Verfügung stellen, um die komplette Information zu erhalten; die Länge des größeren Ausgabebereiches ist dem zweiten Wort des Verwaltungs-Headers zu entnehmen),

– die restlichen 4 Byte werden nicht verwendet.

Der **spezifizierte Ausgabebereich** kann die folgenden Informationen enthalten:

Im vom Aufrufer angegebenen Ausgabebereich wird ein ISAM-Pool beschrieben durch einen sog. ISAM-Pool-Deskriptor in der Länge von 32 Byte.

– den Namen des betroffenen ISAM-Pools in der Länge von 8 Byte (abdruckbar),

– die CATID des Host-Rechners, auf dem der betroffene ISAM-Pool vorhanden ist, ebenfalls abdruckbar in der Länge von 4 Byte,

– die Größe des betroffenen ISAM-Pools (in Einheiten von 2K) in der Länge von 4 Byte,

– den SCOPE (Gültigkeits-Bereich) des betroffenen ISAM-Pools; für diese Information genügt 1 Byte, wobei gilt:

  X'00' -- SCOPE = *TASK
  X'01' -- SCOPE = *USERID
  X'02' -- SCOPE = *HOST
  X'03' -- SCOPE = *USERGROUP

– die WROUT-Eigenschaft des betroffenen ISAM-Pools, wobei die folgenden Werte möglich sind (1 Byte):

  X'00' -- WROUT = DEFERRED
  X'01' -- WROUT = IMMEDIATE

– die Eigenschaft CSTAT-Residenz des betroffenen ISAM-Pools (1 Byte), wobei gilt:

  X'00' -- ISAM-Pool ist nicht CSTAT-resident
  X'01' -- ISAM-Pool ist CSTAT-resident

– die Angabe, ob der betroffene ISAM-Pool einen 2K-Extent, einen 4K-Extent bzw. beide Extents besitzt.
  Diese Angabe ist in einem Byte enthalten mit der Zuordnung:

  X'00' -- noch kein Extent angelegt
  X'01' -- 2K-Extent für ISAM-Pool vorhanden
  X'02' -- 4K-Extent für ISAM-Pool vorhanden
  X'03' -- der ISAM-Pool besitzt sowohl einen 2K- als auch einen 4K-Extent

– Die Angabe, ob der betroffene ISAM-Pool zu einem lokalen oder zu einem Remote-Host-Rechner gehört mit der folgenden Zuordnung:

  X'00' -- Pool auf lokalem Host-Rechner

  X'01' -- Pool auf remote Host-Rechner

– Die USERID, falls der betroffene ISAM-Pool mit dem Attribut SCOPE = USERID versehen ist,

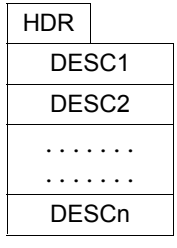
– die restlichen 3 Byte werden nicht verwendet.

Die Strukturen Verwaltungsheader (HDR) und ISAM-Pool-Deskriptor (DESC) bestimmen im Wesentlichen das Format des Ausgabebereiches. Der Aufbau des Ausgabebereiches ist jedoch auch abhängig vom Informationsumfang, der beim Aufruf des Makros SHOPOOL spezifiziert wurde.

INFO = *ATTR

Im Falle INFO=*ATTR enthält der im Verwaltungs-Header des Ausgabebereiches dafür vorgesehene Indikator den Wert X'00', und nach dem Verwaltungs-Header folgen die ISAM-Pool Deskriptoren lückenlos hintereinander; ihre Anzahl ist ebenfalls im Verwaltungs-Header hinterlegt.

Hat der Aufrufer beim Makro-Aufruf den Parameter INFO=*ATTR angegeben (bzw. durch Nicht-Angabe des Parameters diesen Standardwert gewählt), so hat der Ausgabebereich des Anwenders den folgenden Aufbau:



INFO=*ALL

Hat der Aufrufer des Makros SHOPOOL den Parameter INFO=*ALL spezifiziert, so will er in seinem Ausgabebereich neben den statischen Eigenschaften der spezifizierten ISAM-Pools auch die TSN's der Tasks vorfinden, die einen Anschluss zu diesen spezifizierten ISAM-Pools besitzen. angeschlossen sind. Der im Verwaltungs-Header des Ausgabebereiches vorgesehene Indikator für den Funktions-Umfang enthält dann den Wert X'01', und nach jedem ISAM-Pool-Deskriptor folgt ein Wort, das die Anzahl der an diesen ISAM-Pool konnektierten Tasks enthält, sowie im Anschluss daran eine Aufzählung der TSN's der entsprechenden Tasks (jeweils in der Länge 4 Byte, abdruckbar).

Hat der Aufrufer beim Makro-Aufruf den Parameter INFO=*ALL spezifiziert, so hat der Ausgabebereich des Aufrufers den folgenden Aufbau:

HDR					
DESC1					
TASK#1	TSN11	TSN12	. . . . .	TSN1m	
DESC2					
TASK#2	TSN21	TSN22	. . . . .	. . . . .	TSN2r
. . . .	. . . . .				
. . . .	. . . . .				
DESCn					
TASK#n	TSNn1	. . . . .	TSNns		

SHOWAIX – Informationen über Sekundärschlüssel ausgeben

Makrotyp: S-Typ (E-Form/L-Form/D-Form/C-Form) (siehe Seite 870)

Der Makro SHOWAIX übergibt in seiner Operandenliste Informationen über die Sekundärschlüssel einer NK-ISAM-Datei. Zu jedem in der Datei definierten Sekundärschlüssel informiert er den Anwender über

- den Namen des Schlüssels
- die Position des Schlüssels im Datensatz
- die Länge des Schlüssels
- die DUPKEY-Eigenschaft des Schlüssels (mehrfach auftretende Schlüsselwerte erlaubt oder nicht erlaubt)
- die Vollständigkeit der zum Schlüssel gehörenden Sekundärindexblöcke.

Format

Operation	Operanden
SHOWAIX	MF=L,{ FILE=pfadname / LINK=linkname }
	MF=E,PARAM={ adr / (r) }
	MF=D[,PREFIX=pre]
	MF=C[,PREFIX=pre][,MACID=macid]

Operandenbeschreibung

FILE = pfadname
Bezeichnet die NK-ISAM-Datei, über deren Sekundärschlüssel Informationen ausgegeben werden sollen, mit <c-string 1..54: filename 1..54>. Die Angabe im FILE-Operanden wird ignoriert, wenn zugleich der LINK-Operand angegeben ist.

Pfadname bedeutet [:catid:][$userid.]dateiname

catid
Katalogkennung: falls nicht angegeben, wird die Default-Catid der Benutzerkennung angenommen.

userid
Benutzerkennung: falls nicht angegeben, wird die Benutzerkennung des SET-LOGON-PARAMETERS- bzw. des LOGON-Kommandos angenommen.

dateiname
vollqualifizierter Dateiname

LINK = linkname <1..8>
Legt den Dateikettungsnamen der Datei fest, über deren Sekundärschlüssel Informationen ausgegeben werden sollen.
Soll der Dateikettungsname über die Kommandoschnittstelle ansprechbar sein, muss er dem Datentyp <structured_name 1..8> entsprechen (siehe Handbuch „Kommandos“ [3]).

MACID
legt jeweils das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: MACID = ISI

= macid
„macid“ ist eine drei Zeichen lange Zeichenfolge, die jeweils das zweite bis vierte Zeichen der generierten Feldnamen und Equates festlegt.

PARAM
bezeichnet die Adresse der Operandenliste. Der Operand wird nur in Verbindung mit MF=E ausgewertet (siehe auch Seite 870).

= adr
adr ist die symbolische Adresse (der Name) der Operandenliste.

= (r)
r ist die Nummer des Registers, das die Adresse der Operandenliste enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.

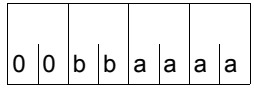
PREFIX
legt das jeweils erste Zeichen der Feldnamen und Equates fest, die bei der Makroauflösung generiert werden.

Voreinstellung: PREFIX = D

= pre
„pre“ ist ein Zeichen langes Präfix, mit dem die generierten Feldnamen und Equates beginnen sollen.

Returncodes

Standard-header:



Über die Ausführung des Makros SHOWAIX wird im Standardheader folgender Returncode übergeben (bb = SUBCODE1, aaaa = MAINCODE):

X'bb'	X'aaaa'	Erläuterung
X'00'	X'0000'	Funktion erfolgreich ausgeführt.
X'01'	X'0001'	Die Operandenliste ist nicht verfügbar.
X'40'	X'0002'	Sekundärschlüssel werden im fernen System nicht unterstützt (bei Makroaufruf über RFA).
X'40'	X'0003'	Die angegebene Katalogkennung existiert nicht.
X'40'	X'0004'	Auf den Katalog kann nicht zugegriffen werden.
X'01'	X'0005'	Die Operandenliste enthält einen ungültigen Namen.
X'20'	X'000B'	Systemfehler.
X'40'	X'000E'	Der Kontrollblock der Datei ist fehlerhaft.
X'01'	X'0017'	In der Operandenliste wurde keine Datei spezifiziert.
X'40'	X'0019'	Der Dateikettungsname ist fehlerhaft.
X'40'	X'0040'	OPEN-Fehler.
X'40'	X'0041'	CLOSE-Fehler.
X'40'	X'0044'	Die Datei ist keine NK-ISAM-Datei.

Weitere Returncodes, deren Bedeutung durch Konvention makroübergreifend festgelegt ist, können der Tabelle auf Seite 873 (Standardheader) entnommen werden.

Das aufrufende Programm wird beendet, wenn bezüglich der Parameterliste folgende Fehler auftreten:

– Die Liste ist dem Aufrufer nicht zugewiesen.
– Die Liste ist nicht auf Wortgrenze ausgerichtet.
– Die Liste ist gegen Schreibzugriff geschützt.

Layout der Operandenliste

Makroauflösung mit MF=D und Standardwerten für PREFIX und MACID:

```
SHOWAIX MF=D
1          MFCHK MF=D,                                                  C
1                PREFIX=D,                                              C
1                MACID=ISI,                                             C
1                DMACID=ISI,                                            C
1                DNAME=ISIAIX,                                          C
1                SUPPORT=(C,D,L,E),                                     C
1                PARAM=,                                                C
1                ALIGN=F,                                               C
1                SVC=32
2 DISIAIX  DSECT ,
2                *,##### PREFIX=D, MACID=ISI #####
1          #INTF INTNAME=SHOWAIX,REFTYPE=REQUEST,INTCOMP=001
1          FHDR  MF=(C,DISI),EQUATES=NO
2          DS    0A
2 DISIFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
2 *
2 DISIIFID DS    0A              0   INTERFACE IDENTIFIER
2 DISIFCTU DS    AL2             0   FUNCTION UNIT NUMBER
2 *                                  BIT 15    HEADER FLAG BIT,
2 *                                  MUST BE RESET UNTIL FURTHER NOTICE
2 *                                  BIT 14-12 UNUSED, MUST BE RESET
2 *                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
2 DISIFCT  DS    AL1             2   FUNCTION NUMBER
2 DISIFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
2 *
2 DISIRET  DS    0A              4   GENERAL RETURN CODE
2 DISISRET DS    0AL2            4   SUB RETURN CODE
2 DISISR2  DS    AL1             4   SUB RETURN CODE 2
2 DISISR1  DS    AL1             5   SUB RETURN CODE 1
2 DISIMRET DS    0AL2            6   MAIN RETURN CODE
2 DISIMR2  DS    AL1             6   MAIN RETURN CODE 2
2 DISIMR1  DS    AL1             7   MAIN RETURN CODE 1
2 DISIFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *
1 * SUB RETURN CODE1
1 *
1 DISIRFSP EQU   X'00'              FUNCTION SUCCESSFULLY PROCESSED
1 DISIRPER EQU   X'01'              PARAMETER SYNTAX ERROR
1 *
1 DISIRFNS EQU   X'01'              CALLED FUNCTION NOT SUPPORTED
1 DISIRFNA EQU   X'02'              CALLED FUNCTION NOT AVAILABLE
1 DISIRVNA EQU   X'03'              INTERFACE VERSION NOT SUPPORTED
```

```
1 *
1 DISIRIER EQU   X'20'             INTERNAL ERROR
1 DISIRCAR EQU   X'40'             CORRECT AND RETRY
1 *
1 * MAIN RETURN CODE
1 *
1 DISIOK   EQU   0                 AIX DELETED
1 DISINPAR EQU   1                 PARLIST NOT ACCESSIBLE
1 DISINREM EQU   2                 NO SUPPORT ON REMOTE HOST
1 DISINCAT EQU   3                 CATID NOT KNOWN
1 DISINACC EQU   4                 CATALOG NOT ACCESSIBLE
1 DISIINVN EQU   5                 INVALID NAME
1 DISISYSE EQU   11                SYSTEM ERROR
1 DISISPAC EQU   12                NO ADDRESS SPACE
1 DISIWRCB EQU   14                WRONG CONTROLBLOCK
1 DISIFNSP EQU   23                FILE NOT SPECIFIED
1 DISILNKE EQU   25                LINKNAME ERROR
1 DISIINOP EQU   31                SSTA INOP
1 DISISSER EQU   32                SSTA INTERNAL ERROR
1 DISIMEMR EQU   33                SSTA MEMORY ERROR
1 DISIOPSE EQU   34                SSTA OPS ERROR
1 DISIOPME EQU   35                SSTA OPS MEMORY ERROR
1 DISIOPER EQU   64                FILE OPEN ERROR
1 DISICLER EQU   65                FILE CLOSE ERROR
1 DISINNKF EQU   68                NO NK-ISAM FILE
1 DISIRLNK EQU   X'FFFF'           LINKAGE ERROR
1 *
1 DISIDMSC DS    AL2               DMSCODE
1 DISIFILE DS    CL54              FILE
1 DISILINK DS    CL8               LINK
1 DISIKEY# DS    H                 NUMBER OF KEYS
1 *
1 DISIKNAM DS    CL8               KEYNAME
1 DISIKPOS DS    H                 KEYPOS
1 DISIKLEN DS    AL1               KEYLEN
1 DISIIND  DS    XL1               INDICATOR
1 DISIDUPK EQU   X'80'             SET:   DUPKEY = YES
1 *                                RESET: DUPKEY = NO
1 DISIINCO EQU   X'40'             SET:   SIX IS INCOMPLETE
1 *                                RESET: SIX IS COMPLETE
1          DS    29CL12
1 DISI#    EQU   (*-DISIFHE)        LENGTH OF STRUCTURE
           END
```

STORE – Satz speichern

Makrotyp: R bei PARMOD=24
0 bei PARMOD=31

Der STORE-Makroaufruf überträgt einen Satz aus dem Anwenderbereich in die Datei, und zwar an die durch seinen Schlüssel definierte Stelle.

Anders als beim INSRT-Makro können mit STORE Sätze mit gleichen Schlüsseln verarbeitet werden. Gilt DUPEKY=YES wird der neue Satz hinter den letzten Satz gleichen Schlüssels geschrieben. Gilt nicht DUPEKY=YES, überschreibt der neue Satz den in der Datei enthaltenen Satz gleichen Schlüssels.

Beim sequenziellen Erstellen/Erweitern einer Datei mit STORE wird der PAD-Faktor nicht berücksichtigt. Im Gegensatz zum PUT-Makroaufruf hat die sequenzielle Anwendung des STORE-Makroaufrufs eine hohe Blocksplittingrate zur Folge und die Datenblöcke sind nur zu ca. 50 % gefüllt.

Format

Operation	Operanden
STORE	$\left\{\begin{array}{l}\texttt{fcbadr}\\\texttt{(1)}\end{array}\right\},\left\{\begin{array}{l}\texttt{area}\\\texttt{(0)}\end{array}\right\}[,\texttt{PARMOD=}\left\{\begin{array}{l}\texttt{24}\\\texttt{31}\end{array}\right\}]$

Operandenbeschreibung

fcbadr
Adresse des FCB für die zu verarbeitende Datei

(1)
Die FCB-Adresse steht im Register 1.

area
Adresse des Satzes, der in die Datei geschreiben werden soll; auch im Locate-Mode muss der zu schreibende Satz an der Adresse „area“ bereitgestellt werden.

(0)
Die Adresse des neuen Satzes steht im Register 0.

PARMOD
Gibt den Generierungsmodus an.

Voreinstellung: der durch den Assembler oder den GPARMOD-Makroaufruf im Programm eingestellte Wert

= 24
Es wird ein Objekt erzeugt, das nur im 16-MB-Adressraum ablauffähig ist (nur 24-Bit-Adressierung).

= 31
Es wird ein Objekt erzeugt, das im 2-GB-Adressraum ablauffähig ist (24-Bit- oder 31-Bit-Adressierung).

Hinweis zur Programmierung

Der STORE-Makroaufruf zerstört die Register 0, 1, 14 und 15.

VERIF – Datei wiederherstellen

Makrotyp: S-Typ (E-Form/L-Form) (siehe Seite 870)

Der VERIF-Makroaufruf dient dazu, eine Datei, Dateigeneration oder Dateigenerationsgruppe, die wegen eines Systemzusammenbruchs oder Auftragsabbruchs nicht ordnungsgemäß geschlossen wurde, wieder verfügbar zu machen.

Mithilfe des Makroaufrufs lässt sich:

– eine Dateisperre aufheben, sodass die Datei wieder allgemein zugänglich wird;
– eine Plattendatei wiederherstellen: der Katalogeintrag wird aktualisiert, wenn nötig, die Datei geschlossen; ISAM-Dateien werden anhand der vorhandenen Datensätze rekonstruiert.

Hinweis

Wurde der Dateizugriff unterbrochen, während sich Datenpuffer im Arbeitsspeicher befanden, können die letzten vorgenommenen Änderungen bei der rekonstruierten Datei fehlen, da der Pufferinhalt erst dann auf den externen Speicher gebracht wird, wenn der Puffer voll ist.

Der Benutzer kann Banddateien und Plattendateien, die mit dem Kommando SECURE-RESOURCE-ALLOCATION exklusiv reserviert wurden, entsperren, falls der Auftrag, der die Sperre verursachte, vom System mit der Konsolmeldung „`Task Pended Indefinitely`“ abgebrochen wurde.

Plattendateien, deren Dateisperre nicht durch das Kommando SECURE-RESOURCE-ALLOCATION verursacht wurde, können nur vom Systemverwalter entsperrt werden.

Bei der Rekonstruktion einer verschlüsselten ISAM-Datei muss das zugehörige Crypto-Kennwort vorgegeben werden.

Format

Operation	Operanden
VERIF	pfadname$_1$[,pfadname$_2$][,MF=L] [,REPAIR={ YES / ABS / NO / CHECK }]
	MF=(E,{ addr / (r) })

Operandenbeschreibung

pfadname1
bezeichnet den Pfadnamen der wiederherzustellenden permanenten oder temporären Datei, Dateigenerationsgruppe oder Dateigenerationen mit:
<c-string 1..54: filename 1..54>

„pfadname1“ bedeutet [:catid:][$userid.]dateiname

catid
Katalogkennung;
Default-Catid: die der Benutzerkennung zugeordnete Katalogkennung

userid
Benutzerkennung;
Default-Userid: die Benutzerkennung des SET-LOGON-PARAMETERS- bzw. LOGON-Kommandos

dateiname
vollqualifizierter Dateiname einer permanenten oder temporären Datei, Dateigeneration oder FGG

pfadname2
bezeichnet die Datei, in der die ISAM-Datei „pfadname1“ rekonstruiert werden soll, mit:
<c-string 1..54: filename 1..54>

„pfadname2“ ist nur bei der Rekonstruktion von ISAM-Dateien sinnvoll und darf nicht mit „pfadname1“ übereinstimmen. „pfadname2“ kann der Name einer permanenten oder temporären Datei bzw. Dateigeneration sein, nicht jedoch der Name einer Dateigenerationsgruppe. Ohne „pfadname2“ erstellt das System eine Arbeitsdatei für die Rekonstruktion der ISAM-Datei. „pfadname2“ muss jedoch angegeben werden, wenn die ISAM-Datei auf Privatplatte gespeichert ist, mit Index- und Datenteil auf verschiedenen Platten. Wenn eine verschlüsselte Datei wiederhergestellt werden muss, erhält „pfadname2“ die gleichen Verschlüsselungsattribute wie „pfadname1“.

„pfadname2“ bedeutet [:catid:][$userid.]dateiname

catid
Katalogkennung;
Default-Catid: die der Benutzerkennung zugeordnete Katalogkennung

userid
Benutzerkennung;
Default-Userid: die Benutzerkennung des SET-LOGON-PARAMETERS- bzw. LOGON-Kommandos

dateiname
vollqualifizierter Dateiname einer permanenten oder temporären Datei oder einer Dateigeneration. Eine Dateigenerationsgruppe kann nicht angegeben werden.

REPAIR
Gibt an, wie die mit „pfadname1“ bezeichnete Datei zu rekonstruieren ist; die Rekonstruktion ist abhängig von der Zugriffsmethode, mit der die Datei erstellt wurde.

„Eine Dateisperre aufheben“ bedeutet: der betreffende Eintrag in der Tabelle, in der die gesperrten Dateien erfasst sind (File Lock Table), wird gelöscht.

Concurrent Copy Locks bleiben bestehen, falls die Concurrent Copy Session noch nicht beendet ist. Bei REPAIR=YES/ABS/CHECK wird eine geforderte Rekonstruktion bzw. ein geforderter Konsistenzcheck trotzdem durchgeführt. Für REPAIR=NO siehe Operandenbeschreibung unter „Hinweise“.

Für Banddateien und Dateigenerationsgruppen ist nur REPAIR=NO möglich.

= YES

Nur für Plattendateien

PAM:
Der Zeiger zur letzten beschriebenen Seite (LPP) wird auf den höchst möglichen Wert gesetzt. Dies ist bei BLKCTRL=PAMKEY die Dateigröße (FILESIZE) und bei BLKCTRL=DATA/NO die Dateigröße abgerundet auf ein Vielfaches der Blockgröße. Dabei wird auch der Last Page Pointer auf Blockgrenze hochgesetzt.

Die Datei wird geschlossen.

Es wird eine Egalisierung der Datei bzgl. einer eventuell vorhandenen Spiegelplatte (DRV) durchgeführt.

SAM:
Die höchste beschriebene PAM-Seite der Datei wird ermittelt, und der Zeiger zur letzten beschriebene Seite (LPP) auf diesen Wert gesetzt.

Befindet sich die Datei auf einer Spiegelplatte (DRV), so wird, falls nötig, die Gleichheit der Blockinhalte durch Egalisierung wiederhergestellt.

Die Datei wird geschlossen.

ISAM:
Die Dateisperre wird aufgehoben und die Datei, falls sie als offen gekennzeichnet ist, rekonstruiert.

= ABS

Nur für Plattendateien:
Die Wiederherstellung wird durchgeführt, unabhängig davon, ob die Datei als offen gekennzeichnet ist oder nicht.

PAM:
Die Dateisperre wird aufgehoben. Anschließend wird der Zeiger zur letzten PAM-Seite (Last Page Pointer) auf die höchste beschriebene PAM-Seite und der Last Byte Pointer auf Blockgrenze hochgesetzt, sofern die Datei als offen gekennzeichnet ist; andernfalls bleiben der Last Page Pointer und Last Byte Pointer unverändert.

Für Dateien mit BLKCTRL=PAMKEY/DATA wird die höchste beschriebene PAM-Seite der Datei ermittelt, und der Zeiger zur letzten beschriebene Seite (LPP) auf diesen Wert gesetzt.

Für Dateien mit BLKCTRL=NO wird der Zeiger zur letzten Seite (LPP) auf die höchst mögliche PAM-Seite (Dateigröße abgerundet auf ein Vielfaches der Blockgröße) gesetzt.

Ist die Datei offen und befindet sie sich auf einer Spiegelplatte (DRV), so wird, falls nötig, die Gleichheit der Blockinhalte durch Egalisierung wiederhergestellt.

Die Datei wird ggf. geschlossen.

SAM:
Die Dateisperre wird aufgehoben; auch wenn die Datei nicht als offen gekennzeichnet ist, wird der Last Page Pointer auf die höchste beschriebene PAM-Seite gesetzt; die Datei wird ggf. geschlossen.

ISAM:
Die Dateisperre wird aufgehoben und die Datei rekonstruiert.

Für die Zugriffsmethoden SAM und ISAM ist die Wiederherstellung analog REPAIR=YES.

= CHECK
nur sinnvoll für NK-ISAM-Dateien, die mit WROUT=YES bearbeitet werden:
es werden nur Dateien ausgewählt, die als offen gekennzeichnet sind. Die Dateisperre wird aufgehoben, der Zeiger zur letzten PAM-Seite wird auf die höchste beschriebene Seite gesetzt, Multiblöcke werden auf Konsistenz der Blockungsstruktur geprüft, die Datei wird geschlossen.

Die Dateisperre wird aufgehoben; ist die Datei als offen gekennzeichnet so wird der Zeiger zum letzten PAM-Block auf die höchste beschriebene Seite gesetzt, die Multiblöcke werden auf Konsistenz geprueft, die vereinbarten Sekundärschlüssel werden auf vollständiges Erzeugen oder Löschen geprüft.

Befindet sich die Datei auf einer Spiegelplatte (DRV), so wird, falls nötig, die Gleichheit der Blockinhalte durch Egalisierung wiederhergestellt.

Wurden keine Fehler erkannt, so wird die Datei geschlossen.

„Konsistenz der Multiblöcke“ bedeutet, es ist kein Abbruch während des Schreibens eines Multiblocks aufgetreten ist.

= NO

für Band-Eingabedateien:
„dateiname“ muss vollqualifiziert angegeben werden; die Dateisperre wird aufgehoben.

für Plattendateien:
PAM – Die Dateisperre wird aufgehoben; die Datei gilt noch nicht als geschlossen, d.h. bei FSTAT ...,STATE=NOCLOS wird sie weiterhin gemeldet, für VERIF ...,REPAIR=YES gilt sie als zu reparierende Datei.

SAM:
Die Dateisperre wird aufgehoben; die Datei gilt noch nicht als geschlossen, d.h. bei FSTAT ...,STATE=NOCLOS wird sie weiterhin gemeldet, für VERIF ...,REPAIR=YES gilt sie als zu reparierende Datei.

ISAM:
Die Dateisperre wird aufgehoben; falls die Datei als offen gekennzeichnet ist, führt das System eine privilegierte Schließoperation durch; der Last Page Pointer wird auf die höchste beschriebene PAM-Seite gesetzt, die Datei wird nicht rekonstruiert.

Befindet sich die Datei auf einer Spiegelplatte (DRV), so wird, falls nötig, die Gleichheit der Blockinhalte durch Egalisierung wiederhergestellt.

Inkonsistenzen zwischen INDEX- und Datenteil werden hierbei weder erkannt noch behoben.

Das gleiche gilt für Inkonsistenzen bezüglich der Sekundärschlüssel. Die Datei gilt jedoch als geschlossen, d.h. bei FSTAT ...,STATE=NOCLOS wird sie nicht gemeldet, für VERIFY ...,REPAIR=YES gilt sie nicht als zu reparierende Datei.

Hinweis in Zusammenhang mit dem Sicherungsverfahren „Concurrent Copy“

Concurrent Copy Locks auf eine Datei werden vom System/HSMS gesetzt (siehe Handbuch „HSMS“ [10]). Diese Locks kann ein Benutzer nur dann zurücksetzen, wenn die Sicherung (Concurrent Copy Session) beendet ist. Hinsichtlich der Returncodes ist hierbei Folgendes zu beachten:

– Ist eine Datei sowohl durch Dateilocks als auch durch Concurrent Copy Locks gesperrt, so wird ein positiver Returncode gesetzt, falls die Dateilocks zurückgesetzt werden konnten.
– Ist eine Datei dagegen nur durch einen Concurrent Copy Lock gesperrt, wird ein positiver Returncode nur dann gesetzt, wenn die Concurrent Copy Session beendet ist, andernfalls wird ein negativer Returncode gesetzt.

MF
Eine ausführliche Beschreibung des MF-Operanden ist im Anhang (Seite 870) enthalten.

Hinweise zur Programmierung

Bei ordnungsgemäßem Abschluss des Makroaufrufs wird der Inhalt des Registers 15 auf null gesetzt. Der Fehlerschlüssel bei nicht ordnungsgemäßem Abschluss ist im IDEMS-Makroaufruf definiert.

Rekonstruktion von ISAM-Dateien

– Fehlt die Angabe „pfadname2“ für eine ISAM-Datei auf gemeinschaftlichen Datenträgern, wird sie in einer Arbeitsdatei rekonstruiert, die vom System erstellt wird. Anschließend wird die Datei „pfadname1“ gelöscht, und zwar ohne „DESTROY“ (siehe Makro CATAL, Operand DESTROY, Seite 153), und die Arbeitsdatei in „pfadname1“ umbenannt.

  Fehlt die Angabe „pfadname2“ für eine ISAM-Datei auf privaten Datenträgern oder auf einem Net-Storage-Volume, wird sie in einer temporären Arbeitsdatei auf gemeinschaftlichen Datenträgern rekonstruiert. Anschließend wird die Arbeitsdatei in die Datei „pfadname1“ kopiert und mit „DESTROY“ (siehe Makro CATAL, Operand DESTROY, Seite 153) gelöscht. Dieser Vorgang kann sehr zeitaufwändig sein, sodass es günstiger ist, „pfadname2“ anzugeben.

– Wird im VERIF-Makroaufruf „pfadname2“ angegeben, wird „pfadname1“ dort rekonstruiert. „pfadname1“ selbst bleibt unverändert. Soll „pfadname2“ auf privaten Datenträgern oder auf einem Net-Storage-Volume stehen oder handelt es sich bei „pfadname1“ um eine ISAM-Datei mit Daten- und Indexteil auf getrennten Privatplatten, muss „pfadname2“ vor Eingabe des VERIF-Makroaufrufs katalogisiert und Speicherplatz reserviert worden sein.

  Datensätze, bei denen Index und Daten gleich sind, werden nur einmal in die rekonstruierte Datei übernommen.

– In den Datenblöcken der rekonstruierten Datei wird kein Platz für spätere Erweiterungen frei gehalten, was der Vereinbarung PAD=0 im FILE-Makroaufruf entspricht.

  ISAM-Dateien mit Daten- und Indexblöcken auf getrennten privaten Datenträgern können mit dem VERIF-Makroaufruf nur rekonstruiert werden, wenn BLKSIZE=STD gilt.

– Enthält ein ISAM-Datenblock Daten, die keinem definierten Datensatz zugeordnet werden können, wird der gesamte Block in der PAM-Datei „S.dateiname1.REPAIR“ sichergestellt. Nach der VERIF-Bearbeitung steht diese Datei dem Benutzer zu eigenen Rekonstruktionsversuchen zur Verfügung. Falls der neue Dateiname zu lang wird, wird „dateiname1“ entsprechend gekürzt.

– Da bei der Wiederherstellung von ISAM-Dateien eine Dateikopie angelegt wird, die zum Pubspace zählt, muss der Anwender dafür sorgen, dass ihm genügend Speicherplatz zur Verfügung steht.

5 Anhang

Im Anhang finden Sie:

- Informationen zur Syntaxdarstellung der Makroaufrufe (ab Seite 862)
- einen Abschnitt zur Auswertung von DVS-Fehlermeldungen (ab Seite 875)
- die CALL-Schnitstelle für DIV (ab Seite 880)
- alle Kennsatzformate einschließlich Informationen zur Verarbeitung der Kennsatzfelder (ab Seite 883)
- Informationen zur Erzeugung einer Dsect (ab Seite 899)

Den Abschluss bilden die Makroformate der zwei ersetzten Makros COPY und REL, die aus Kompatibilitätsgründen noch unterstützt werden (ab Seite 901).

5.1 Syntaxdarstellung

5.1.1 Makroaufrufformat

Das Makroaufrufformat ist aus zwei Spalten aufgebaut; die erste Spalte enthält den Makronamen, die zweite Spalte die möglichen Operanden.

Makroname	Operanden
<makroname>	<operand $_1$> ,<operand $_2$>

Beim Aufruf des Makros muss der Makroname mit mindestens einer Leerspalte vom ersten Operanden getrennt sein. Mehrere Operanden müssen durch Kommata getrennt angegeben werden.

Im Makroaufrufformat werden bestimmte Zeichen (Metazeichen) verwendet, die in den folgenden Tabellen erläutert werden.

5.1.2 Metasyntax der Makroaufrufformate

Elemente der Metasyntax

Kennzeichnung	Bedeutung	Beispiele
GROSSBUCHSTABEN	Großbuchstaben bezeichnen Schlüsselwörter oder Konstanten, die in dieser Form vom Benutzer angegeben werden müssen. Schlüsselwörter müssen mit * beginnen, falls alternativ sowohl Schlüsselwörter als auch Namen von Konstanten oder Variablen angegeben werden können.	DIB FORCED=*YES
Kleinbuchstaben	Kleinbuchstaben bezeichnen Datentypen der Werte oder Variablen, die vom Benutzer angegeben werden können.	DIB = <var: pointer>
< >	Spitze Klammern kennzeichnen Variablen, deren Wertevorrat durch die Datentypen beschrieben wird.	<var: pointer>
Unterstreichung	Der Unterstrich kennzeichnet den Default-Wert eines Operanden. Hat ein Operand keinen Default-Wert, so ist die Angabe eines Operanden Pflicht.	FORCED = *NO / *YES
=	Das Gleichheitszeichen verbindet den Operandennamen mit den dazugehörenden Operandenwerten.	DATA = <var: pointer>
/	Der Schrägstrich trennt alternative Operandenwerte.	FORCED = *NO / *YES
list-poss(n)	Aus den list-poss folgenden Operandenwerten kann eine Liste gebildet werden. n gibt die maximale Anzahl der Listenelemente an. Enthält die Liste mehr als ein Element, muss sie in runden Klammern eingeschlossen werden.	FLAG=list-poss(3): *SLI / *SKIP / *DC Angabe: FLAG=*SKIP FLAG=(*SLI,*DC)

Einem Operanden wird durch ein Gleichheitszeichen ein Operandenwert zugewiesen, welcher aus einem definierten Wertevorrat stammt.
Dieser Wertevorrat wird durch einen Datentyp bestimmt. Nachfolgende Tabelle enthält die Datentypen der Operandenwerte.

Datentypen der Operandenwerte

Datentyp	Zeichenvorrat	Anmerkungen
c-string	EBCDIC-Zeichen	ist in Hochkommata einzuschließen
integer	[+-] 0..2147483647	ist eine dezimale Zahl
var:	leitet eine variable Angabe ein. Nach dem Doppelpunkt folgt der Typ der Variablen (siehe Tabelle Datentypen der Variablen)	<var:var-type>
reg:	Register 0..15	Angabe: (<reg:var-type>)

Zusätze zu Datentypen

Zusatz	Bedeutung
n..m	für Datentyp integer bedeutet n..m eine Intervallangabe; n: Mindestwert m: Maximalwert
	für Datentyp c-string bedeutet n..m eine Längenangabe in Byte; n: Mindestlänge m: Maximallänge mit n < m
n	bei Datentyp c-string bedeutet n eine Längenangabe in Byte; n muss exakt eingehalten werden.

Die Operandenwerte können direkt als Zeichenkette oder Integer-Zahl (siehe Datentypen „c-string“ und „integer“) eingegeben werden oder indirekt über eine Variable (siehe Datentyp „var:“) bezeichnet werden. Die nachfolgende Tabelle enthält die Datentypen, die für Variablen möglich sind.

Datentypen der Variablen

Datentyp	Beschreibung	Definition im Programm
char:n	Die Variable ist eine Zeichenkette von n Zeichen. Fehlt die Längenangabe, wird n=1 angenommen.	CLn
int:n	Die Variable ist eine Integer-Zahl, die n Byte belegt. Fehlt die Längenangabe, wird n=1 angenommen. Bedingung: $n \leq 4$	FLn
enum-of E:n	Die Variable ist die Aufzählung E, die n Byte belegt. Fehlt die Längenangabe, wird n=1 angenommen. ($n \leq 4$)	XLn
pointer	Die Variable ist eine Adresse oder ein Adresswert.	A

5.1.3 Veraltete Metasyntax der Makroaufrufformate

Bei der alten Darstellung von Formaten werden bestimmte Zeichen (so genannte Metazeichen) verwendet und Vereinbarungen getroffen, die in der folgenden Tabelle erläutert sind:

Formale Darstellung	**Erläuterung**	**Beispiel**
GROSSBUCH-STABEN	Großbuchstaben bezeichnen Konstanten, die in dieser Form vom Benutzer eingegeben werden müssen.	FSTAT ,LIST=(SYSLST) Einzugeben ist: FSTAT ,LIST=(SYSLST)
Kleinbuchstaben	Kleinbuchstaben bezeichnen Variablen, die bei der Eingabe vom Benutzer durch aktuelle Werte ersetzt werden müssen, d.h. ihr Inhalt kann von Fall zu Fall verschieden sein.	FILE dateiname Einzugeben ist: FILE DATEI, FILE XYZ, FILE A.B-1, usw.
{ }	Geschweifte Klammern schließen Alternativen ein, d.h. aus den eingeschlossenen Größen muss eine Angabe ausgewählt werden.	{ FILE=pfadname LINK=name } einzugeben ist: FILE=... oder LINK=...
[ ]	Eckige Klammern schließen Wahlangaben ein, d.h. Angaben, die man weglassen darf. Steht bei Wahlangaben das Komma innerhalb Klammer, so wird es nur bei Verwendung dieser Wahlangabe verlangt und kann beim ersten Operanden weggelassen werden. Steht es außerhalb der Klammer, so muss es stets geschrieben werden, auch wenn keine Wahlangabe gemacht wird. (Runde Klammern müssen eingegeben werden.)	F[REE]SIZE einzugeben ist FREESIZE oder abgekürzt FSIZE
___	Die Unterstreichung hebt den Standardwert (Voreinstellung) hervor. Das ist der Wert, den das System einsetzt, wenn der Benutzer keine Angabe macht.	[{ ISAM SAM }] Einzugeben ist: SAM oder ISAM oder nichts (= ISAM)

Formale Darstellung	Erläuterung	Beispiel
...	Punkte bedeuten eine Wiederholung. Sie zeigen an, dass die davor stehende Einheit mehrmals hintereinander wiederholt werden kann.	(vsn,...) Einzugeben ist: (PVT003) oder (PVT003,PVT456) oder (XY00AB,XY0012,XY0005) usw.
_	Dieses Zeichen kennzeichnet ein Leerzeichen (X'40').	STD_ Anzugeben ist 'STD '

5.1.4 Musterzeichen

Musterzeichen sind in den Makros ERASE und FSTAT zulässig im Operanden „pfadname“. Der Anwender kann für Katalogkennung und Dateinamen Muster verwenden. Im ERASE-Makro können auch Systemdateien über Muster angesprochen werden.

Muster	**Bedeutung**
*	Ersetzt eine beliebige Zeichenfolge, auch die leere Zeichenfolge.
/	Ersetzt genau ein beliebiges Zeichen.
<muster1,...>	Ersetzt alle Zeichenfolgen, auf die eines der angegebenen Muster zutrifft
<muster1:muster2>	Ersetzt eine Zeichenfolge, für die gilt: – sie ist mindestens so lang wie die kürzeste Muster-Zeichenfolge – sie ist höchstens so lang wie die längste Muster-Zeichenfolge – sie liegt in der alphabetischen Sortierung zwischen „muster1“ und „muster2“; Zahlen werden hinter Buchstaben sortiert „muster1“ darf auch die leere Zeichenfolge sein, die in der alphabetischen Sortierung an erster Stelle steht.
<muster1:muster2,...>	Muster der Art „muster1:muster2“ können auch in Listenform angegeben werden. Für jede derartige Bereichsangabe gelten die oben genannten Regeln. Das System nimmt eine logische Oder-Verknüpfung vor, d.h. die Musterliste ersetzt alle Zeichenfolgen, auf die eine der Bereichsangaben zutrifft. Die Längenmerkmale gelten paarweise, d.h. jeweils für eine Bereichsangabe „muster1:muster2“, nicht für die gesamte Liste.
-muster	Ersetzt alle Zeichenfolgen, die dem angegebenen Muster nicht entsprechen. Das Minuszeichen darf nur am Beginn der Musterzeichenfolge stehen.

5.1.5 Format von Datumsangaben

Datumsangaben werden benötigt in den Makros ERASE und FSTAT, jeweils in den Operanden CRDATE, DELDATE, EXDATE, LADATE und LCDATE, sowie im Makro CATAL in den Operanden DELDATE und EXDATE. Der Anwender kann zwischen absoluten und relativen Datumsangaben wählen.
Zusätzlich zum Datum kann eine Zeit bzw. ein Zeitintervall angegeben werden.

Mit dem Operanden **TIMBASE** kann gesteuert werden, ob die absoluten Datumsangaben auf Basis der Weltzeit **UTC** (universal time coordinate) oder auf Basis der lokalen Zeit **LTI** (local time) erfolgen (relative Angaben beziehen sich stets auf die lokale Zeit).

An diesen Operand ist auch das Datumsformat der FSTAT-Ausgabe gekoppelt.

absolute Datumsangabe

ein konkretes Datum in der Form YYMMDD oder
[YY]YY-[M]M-[D]D (YY = Jahr, MM = Monat, DD = Tag)

relative Datumsangabe

als Distanz zum aktuellen Tagesdatum in der Form -n für die Vergangenheit und +n für die Zukunft (n=0..99999);
als Y[ESTERDAY] (≙ -1), T[ODAY] (≙ ±0)
oder TOM[ORROW] (≙ +1)

Zeitangabe

eine Uhrzeit bezogen auf das Datum in der Form datum(hh:mm:ss)
(hh = Stunden, mm = Minuten, ss = Sekunden)

5.1.6 Typen von Makroaufrufen

In Typen werden die Makroaufrufe abhängig von der Art ihrer Operandenübergabe eingeteilt. Es gibt den O-Typ, R-Typ (Übergabe in Registern) und den S-Typ (Übergabe im Speicher).

O-Typ-Makroaufrufe

Makroaufrufe, die weder dem R-Typ noch dem S-Typ zuzuordnen sind, werden als O-Typ-Makros bezeichnet.

Vom O-Typ sind z.B. jene Makroaufrufe, die im Operandenfeld die Angabe eines Registers vorsehen (häufig nur R1), das die Anfangsadresse einer Operandenliste enthält.

Die Operandenliste wird im Datenteil des Programms definiert (DC-Anweisungen) und enthält die Operandenwerte.

R-Typ-Makroaufrufe

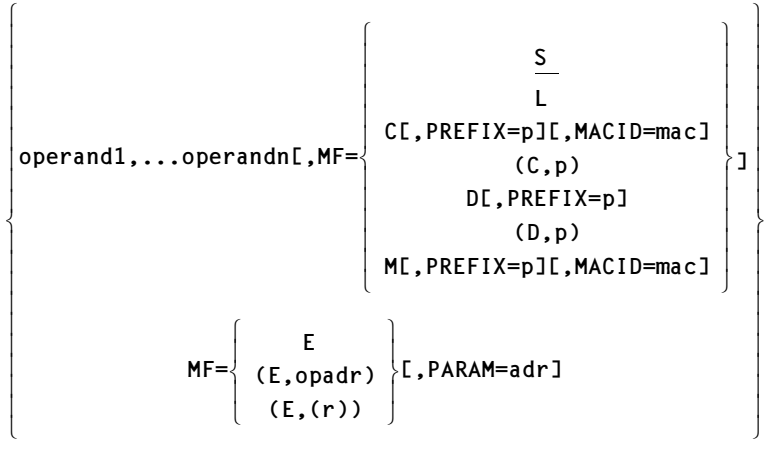
Operation	Operanden
RTYP	{ operand1 | (r1) }, { operand2 | (r2) }

Ein Makroaufruf ist vom Typ R, wenn alle erforderlichen Operandenwerte in die zwei für diesen Zweck verwendeten Register 0 und 1 geladen werden können. Durch einen R-Typ-Makroaufruf wird keine Operandenliste generiert.

Die Operanden können direkt angegeben werden oder in den Registern 0 und 1 enthalten sein.

Adressoperanden in R-Typ-Makroaufrufen können als explizite oder implizite Adressen geschrieben werden.

S-Typ-Makroaufrufe

Name	Operation	Operanden
[opadr]	makro	

Beim S-Typ werden die im Makroaufruf angegebenen Operandenwerte in Form eines Datenbereichs an den Funktionsbaustein übergeben. Der Datenbereich ist Teil der Makroauflösung. Er ist ein geeignet strukturierter Bereich, der die für die Übergabe der Operandenwerte notwendigen Daten- und Speicherdefinitionen (DC- und DS-Anweisungen) enthält.

Für alle Makros, die mit einer bestimmten Makroversion aufgerufen werden können (z.B. über die Operanden VERSION oder PARMOD), gilt: In allen Aufrufen, die sich durch den MF-Operanden unterscheiden (MF=L/E/D/C) muss der Versionsoperand den gleichen Wert haben.

Für MF unterscheidet man 6 Formen des Makroaufrufs: S-Form, E-Form, L-Form, D-Form, C-Form, M-Form.

S-Form = Standardform

MF=S ist Voreinstellung. Es werden zuerst der Befehlsteil und anschließend der Datenbereich generiert, unter Beachtung der im Makroaufruf angegebenen Operandenwerte.
Der Datenbereich enthält keine Feldnamen und keine erläuternden Equates. Der Standardheader ist initialisiert.

E-Form = Execute-Form

	Operation	Operanden
[label]	makro	MF=E,PARAM={addr / (r) / (1)}

Mit der E-Form des MF-Operanden wird ein Systemaufruf (SVC) ausgelöst: der Inhalt einer Operandenliste (siehe L-Form) wird ausgewertet und die entsprechenden Operationen werden ausgeführt. Daher muss im „Execute“-Makroaufruf die Adresse der Operandenliste enthalten sein, entweder als symbolische Adresse (addr) oder in einem Register (r/1). Weitere Operanden werden nicht ausgewertet.

„label“ ist die symbolische Adresse, die dem Makroaufruf im Assemblerprogramm zugeordnet werden kann.

L-Form = LIST-Form

	Operation	Operanden
label	makro	MF=L,operandenliste

Die List-Form erzeugt mit den übrigen im Makroaufruf angegebenen Operanden eine Operandenliste, allerdings ohne symbolische Adressen für die Operanden. Diese werden über die C- oder D-Form generiert. Die Adresse der Operandenliste muss beim Systemaufruf (E-Form) angegeben werden, daher kann die symbolische Adresse „label“ nicht entfallen.

Die Operandenliste beginnt mit dem Standardheader (siehe Seite 873), dessen Felder beim Aufbau der Liste über MF=L automatisch versorgt sind. Auch wenn eine Operandenliste über die D- oder die C-Form dynamisch aufgebaut werden soll, muss sie vorher mit MF=L initialisiert werden, damit eine korrekte Versorgung des Headers gewährleistet ist.

D-Form=DSECT-Form

	Operation	Operanden
[label]	makro	MF=D[,PREFIX=prefix]

Die D-Form erzeugt eine DSECT für die Operandenliste des Makros. Mit dem Operanden PREFIX kann das erste Zeichen der generierten Namen geändert werden. Ist für den Makro mit „label“ eine symbolische Adresse definiert, erhält die DSECT diesen Namen.

Ist „label“ nicht definiert, erhält die DSECT einen makrospezifischen Standardnamen, dessen erstes Zeichen ebenfalls über PREFIX modifiziert wird. Die Operandenliste sollte vor dem DSECT-Aufruf mit der List-Form initialisiert werden, damit die korrekte Versorgung des Standardheaders gewährleistet ist.

C-Form

	Operation	**Operanden**
`[label]`	`makro`	`MF=C[,PREFIX=prefix][,MACID=id]`

Wie die D-Form erzeugt die C-Form eine Operandenliste, allerdings nicht als DSECT, da keine DSECT-Anweisung generiert wird. Die Operandenliste bleibt leer, sie sollte mit einem Aufruf in L-Form initialisiert werden, damit vor allem der Standardheader korrekt versorgt ist.

Mit dem Operanden PREFIX kann das erste Zeichen der generierten Namen geändert werden, mit dem Operanden MACID auch das zweite bis vierte Zeichen (in MACID kann eine bis zu drei Zeichen lange Zeichenfolge angegeben werden). Ist der Makroaufruf mit „label“ symbolisch adressiert, ist dies gleichzeitig die Adresse der Operandenliste; ist „label“ nicht definiert, ist die Operandenliste nicht symbolisch adressierbar.

M-Form = Modifizierungs-Form

	Operation	**Operanden**
[label]	`makro`	`MF=M[,PREFIX=prefix][,MACID=id],operandenliste`

Es werden Befehle (z.B. MVCs) generiert, die während des Programmlaufs in einem bereits initialisierten Datenbereich (Operandenliste) Felder mit den Operandenwerten überschreiben, die im Makroaufruf angegeben werden. Damit bietet die M-Form eine komfortable Möglichkeit, die Operandenwerte, mit denen ein Makro aufgerufen wird, dynamisch dem Programmlauf anzupassen.

Da die dafür generierten Befehle die symbolischen Adressen und Equates der C-Form bzw. D-Form benutzen, ist bei der Verwendung der M-Form sicherzustellen, dass diese Namen für die Adressierung der zu modifizierenden Operandenliste zur Verfügung stehen. Insbesondere ist darauf zu achten, dass bei einem Makroaufruf mit MF=M die ggf. Operanden PREFIX und MACID mit den gleichen Werten angegeben werden wie im zugehörigen MF=C- bzw. MF=D-Aufruf.

5.1.7 Standardheader

Alle DVS-Makros benutzen in ihrer 31-Bit-Schnittstelle den Standardheader für BS2000-Makros. Dieser Standardheader ist ein 8-Byte-Feld am Beginn der Operandenliste, das die normierte Bezeichnung der Schnittstelle enthält und Returncodes aufnehmen kann. Er wird vom jeweiligen Makro erzeugt und sollte – wenn möglich – mit der List-Form der MF-Operanden initialisiert werden.

Aufbau des Standardheaders

Feldinhalt	Byte-Position	Bedeutung
UNIT	0-1	bezeichnet die Funktionseinheit, in der die gesuchte Funktion realisiert ist
FUNCTION	2	bezeichnet die Funktion (innerhalb der Funktionseinheit)
VERSION	3	bezeichnet den „Änderungsstand“, d.h. die Versionsnummer der Funktion
SUBCODE2	4	enthält den Sub-Returncode2
SUBCODE1	5	enthält den Sub-Returncode1
MAINCODE	6-7	enthält den Haupt-Returncode

Die Felder SUBCODE2, SUBCODE1, MAINCODE enthalten den Returncode. Der Haupt-Returncode zeigt an, ob eine Operation erfolgreich ausgeführt werden konnte. Im Fehlerfall kann durch die Sub-Returncodes der Fehler festgestellt werden.

Folgende Werte des Returncodes sind Konvention:

SUBCODE2	SUBCODE1	MAINCODE	Bedeutung
X'00'	X'00'	X'0000'	Erfolgreiche Funktionsausführung. Es gibt keine zusätzlichen Informationen zum MAINCODE.
X'01'	X'00'	X'0000'	Erfolgreiche Funktionsausführung. Es sind keine weiteren Aktionen erforderlich.
X'00'	X'01'	X'FFFF'	Die angeforderte Funktion wird nicht unterstützt (falsche Angabe für UNIT oder FUNCTION im Standardheader). Nicht behebbarer Fehler.
X'00'	X'02'	X'FFFF'	Die angeforderte Funktion ist nicht verfügbar. Nicht behebbarer Fehler.
X'00'	X'03'	X'FFFF'	Die angegebene Version der Schnittstelle wird nicht unterstützt (falsche Versionsangabe im Standardheader). Nicht behebbarer Fehler.
X'00'	X'04'	X'FFFF'	Parameterliste ist nicht auf Wortgrenze ausgerichtet.
X'00'	X'41'	X'FFFF'	Das Subsystem ist nicht vorhanden; es muss explizit erzeugt werden.
X'00'	X'42'	X'FFFF'	Der aufrufende Ablauf ist mit dieser Schnittstelle nicht konnektiert; er muss explizit konnektiert werden.
X'00'	X'81'	X'FFFF'	Subsystem zurzeit nicht verfügbar.
X'00'	X'82'	X'FFFF'	Subsystem im DELETE- oder HOLD-Zustand.

MAINCODE kennzeichnet das Ergebnis der Funktionsausführung. SUBCODE1 dient der Qualifizierung des Hauptwertes. SUBCODE2 dient der weiteren Unterteilung des Fehlers in Fehlerklassen.

Der Returncode sollte ausschließlich im Standardheader übergeben werden (Ausnahmen DVS-Makros siehe Seite 875). In einer Übergangsphase kann der Returncode aber auch im Register R15 oder sowohl im Standardheader als auch im Register R15 übergeben werden. Um zu prüfen, ob im Standardheader ein Returncode übergeben wurde, sollte das Returncode-Feld mit X'FFFFFFFF' vorbesetzt werden. Das Ergebnis der Prüfung des Standardheaders wird in jedem Fall auch im Register R15 übergeben:

X'00000000': Standardheader richtig initialisiert; normale Ausführung.
X'0001FFFF': Falsche Angabe für UNIT oder FUNCTION.
X'0003FFFF': Falsche Angabe für VERSION.

5.2 Fehlermeldungsschlüssel im DVS

Fehlerschlüssel enthalten Informationen über Fehlerzustände von Programmen oder Aufträgen, denen der Anwender Art und Herkunft des Fehlers sowie die zu treffende Maßnahme entnehmen kann. Die DVS-Fehlerschlüssel bieten folgende Vorteile:

– Die kurze Form gestattet die Verschlüsselung vieler Fehlerfälle.

– Aus dem Schlüssel kann die Komponente ermittelt werden, bei der der Fehler aufgetreten ist.

– Bei behebbaren Fehlern kann durch Selbstanalyse eine abnormale Programmbeendigung vermieden werden.

DVS-Fehler beim Programmablauf

Bei folgenden Makroaufrufen der Dateiverwaltung wird der Fehlerschlüssel (mit Subcodes) im Standardheader abgelegt:
– CATAL mit VERSION=3
– COMPFIL
– COPFILE
– DECFILE
– DROPTFT
– ENCFILE
– ERASE mit VERSION=3
– FILE mit VERSION=3
– FSTAT mit VERSION=3/4
– LFFSNAP
– LJFSNAP
– MAILFIL
– RDTFT mit VERSION=3
– RELTFT
– RFFSNAP
– RJFSNAP

Bei folgenden Makroaufrufen der Dateiverwaltung wird der Fehlerschlüssel (ohne Subcodes) im Register 15 in den beiden niedrigerwertigen Byte abgelegt:
– CATAL mit VERSION<3
– CHNGE
– COPY
– ERASE mit VERSION<3
– FILE mit VERSION<3
– FSTAT mit VERSION<3
– IMPORT
– RDTFT ohne VERSION und mit VERSION=2

– REL

Bei Makroaufrufen der Datenverwaltung (OPEN, CLOSE und Zugriffsmethoden) werden Fehlerschlüssel im Feld ID1ECB des TU-FCB (Distanz X'98') abgespeichert.

Tritt während des Programmlaufs ein DVS-Fehler auf, wird ein Fehlerschlüssel im Dateisteuerblock gespeichert. Ist im Programm keine Routine zur Behandlung des aufgetretenen Fehlers vorgesehen, wird das Programm beendet mit Fehlermeldung auf SYSOUT im Dialog- oder auf SYSLST im Batchbetrieb.

Methodik der Verschlüsselung

w	x	y	z

Der Fehlerschlüssel ist eine vierstellige Sedezimalzahl, deren zweite Ziffer x die Komponente bezeichnet, die den Fehler festgestellt hat. Die Ziffern w, y und z bezeichnen den aufgetretenen Fehler.

Fehlercode-Wertetabelle der Komponenten:

x	**Komponente**
2	Privilegiertes PAM (PPAM)
3	Katalogverwalter (CMS)
4	Datenspeicherverwalter (ALLOC)
5 6	} DMS-Kommandos / Assembler-Makros
7	Privilegierte Bandzugriffsmethode (PTAM)
9	UPAM
A	ISAM
B	SAM
C	BTAM
D	OPEN
E	CLOSE

In der Reihenfolge der Komponenten von 2_{16}-E_{16} kommt zum Ausdruck, dass eine Komponente mit höherer Nummer eine Komponente mit niedrigerer Nummer aufrufen kann (und nicht umgekehrt).

Tritt ein Fehler in einer aufgerufenen Komponente auf, wird der Fehlerschlüssel der aufrufenden Komponente übergeben und dabei modifiziert.

Beispiel Generierung und Modifizierung von Fehlerschlüsseln

– *einstufig* (Fehler in der aufrufenden Komponente): Ein Anwender ruft UPAM auf, und die Datei ist nicht eröffnet: UPAM setzt den Fehlerschlüssel 0994 im FCB ab und verzweigt über $GOTO und USERERR.

– *zweistufig* (Fehler in der aufgerufenen Komponente): Ein Anwender ruft UPAM auf, das seinerseits PPAM aufruft, das einen Ein-/Ausgabe-Fehler entdeckt: In diesem Fall gibt PPAM den Fehlerschlüssel 0227 an UPAM im Register 15 weiter. UPAM ändert diesen Schlüssel zu 0927, setzt ihn im FCB ab und verzweigt über $GOTO nach ERRADR.

Liste der DVS-Fehlerschlüssel

Makroaufruf IDEMS

Die Erläuterungen, die den Fehlerschlüsseln zugeordnet sind, stehen in einer Liste. Diese Liste kann ganz oder teilweise ausgegeben werden mithilfe des Makroaufrufs IDEMS.

Operation	Operanden	Bemerkung
IDEMS	[ALL=Y]	alle Fehlerschlüssel werden generiert
	[,PAM=Y]	nur PPAM-Schlüssel
	[,CATAL=Y]	nur CMS-Schlüssel
	[,ALLOC=Y]	nur ALLOC-Schlüssel
	[,CMDMAC=Y]	nur Schlüssel für DMS-Kommandos/Makros Teil 1
	[,CMDNMAC=Y]	nur Schlüssel für DMS-Kommandos/Makros Teil 2
	[,NPAM=Y]	nur UPAM-Schlüssel
	[,ISAM=Y]	nur ISAM-Schlüssel
	[,SAM=Y]	nur SAM-Schlüssel
	[,BTAM=Y]	nur BTAM-Schlüssel
	[,OPEN=Y]	nur Schlüssel der OPEN-Verarbeitung
	[,CLOSE=Y]	nur Schlüssel der CLOSE-Verarbeitung
	[,P=buchstabe]	Präfix für die symbolischen Namen der DVS-Meldungen; Standardwert: I Für „P=“ wird kein Buchstabe generiert.

Präfix der Systemmeldungen

Bei der Ausgabe von Systemmeldungen wird dem Fehlerschlüssel die Meldungsklasse „DMS“ vorangestellt (0D33 → DMS0D33). Mit dem Kommando bzw. der Standardanweisung HELP-MSG-INFORMATION können zu einem Meldungsschlüssel Meldungstext sowie weitere Angaben zu Fehlerursache und -behandlung in deutsch bzw. englisch ausgegeben werden.

Ausnahmen

– ISAM: für ISAM gilt stets als Meldungsschlüssel 0AAz (0 ≤ z ≤ F), auch bei den Systemmeldungen.

  Beispiel
  Wenn PPAM nach ISAM-Aufruf einen Fehler feststellt, wird immer der '0AA9' im FCB eingetragen. Der Eintrag in der IDEMS-Liste lautet „SYSTEM ERROR, HARDWARE“. ISAM gibt diese Meldung für alle in PPAM festgestellten Fehler aus.

– OPEN: Bei Eröffnen von ISAM-Dateien können Fehler auftreten, die nicht über den Fehlerschlüssel analysierbar sind.

  Beispiel
  Wenn eine ISAM-Datei im INPUT-Modus eröffnet wird, wird die „erste PAM-Seite“ gelesen. Sie enthält immer den Kontrollblock (NK-ISAM) oder den höchsten Indexblock (K-ISAM).

  Nach einem Ein-/Ausgabe-Fehler geschieht Folgendes:

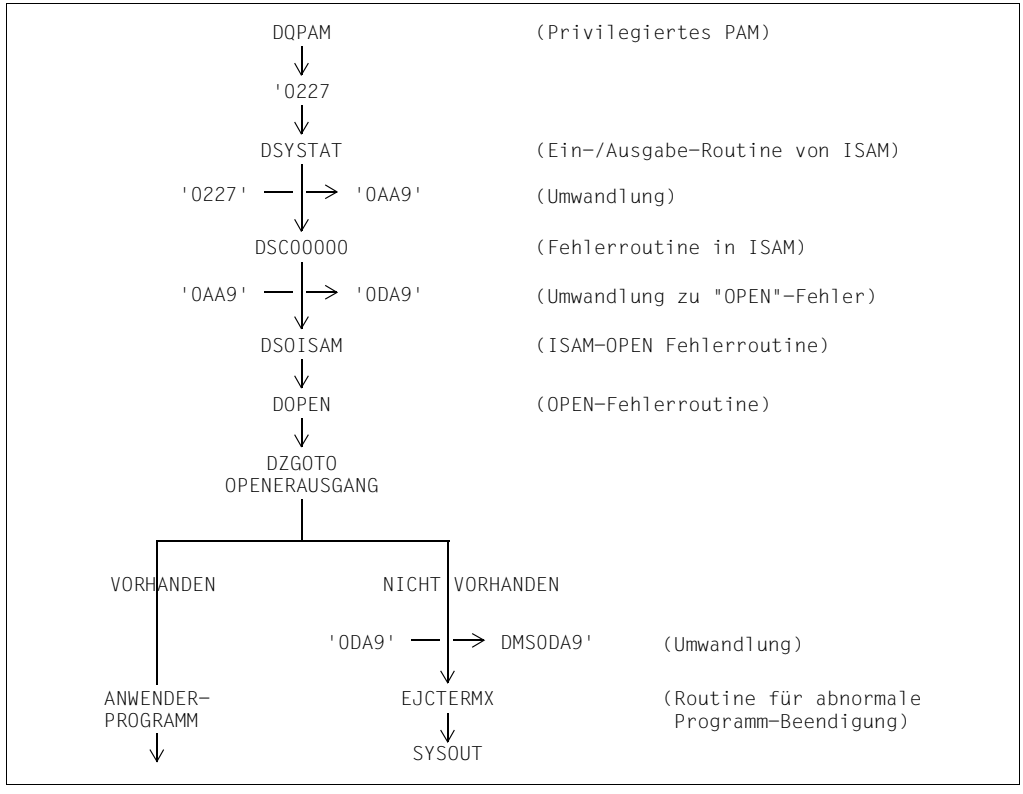


Bild 11: Fehlerablauf nach einem Ein-/Ausgabe-Fehler

Hinweis

Dieser Fehler hat im IDEMS-Makroaufruf eine andere Bedeutung, hier aber steht er für Ein-/Ausgabe-Fehler.

– CLOSE: Um alle noch ausstehenden Ein-/Ausgaben zu beenden, ruft die Routine CLOSE die Zugriffsmethode der Datei auf.

Hilfen für die Fehleranalyse

Zur Analyse eines Fehlerschlüssels gibt es folgende Hilfsmittel:

– Bedeutung in der IDEMS-Liste aufsuchen.

– Werden in einem Programm Makroaufrufe der Dateiverwaltung benutzt, sollte der Anwender den Fehlerschlüssel (im Standardheader bzw. in Register 15) immer überprüfen. Falls der Fehlerschlüssel $\neq 0$ ist, sollte das Programm beendet werden oder eine gründliche Analyse des Fehlerschlüssels vorgenommen werden.

5.3 CALL-Schnittstelle für DIV

Mit dem Adapter Window Services ist es in einigen höheren Programmiersprachen möglich, DIV-Funktionen aus Programmen über eine CALL-Schnittstelle aufzurufen. Es handelt sich dabei um folgende Programmiersprachen:

– COBOL (ab Version 2.0)
– FORTRAN (ab Version 2.2)
– PL/1 (ab Version 4.1)
– C (ab Version 2.0)

Der Aufruf von DIV-Funktionen über eine CALL-Schnittstelle setzt die sog. ILCS-Linkage voraus.
Die Übersetzung entsprechender Programme in den Programmiersprachen COBOL, FORTRAN, und C liefert die ILCS-Linkage standardmäßig. Bei der Übersetzung eines PL/1-Programms muss eine entsprechende Compiler-Option eingestellt werden.

Eine Besonderheit gibt es bei Programmen in C zu beachten: es sind nicht die Datenelemente, sondern Zeiger auf die Datenelemente zu übergeben („call by reference“-Übergabe in C).

Die Semantik der Datentypen weist bei den oben genannten Programmiersprachen starke Unterschiede auf. In der folgenden Tabelle werden diejenigen Datentypen aufgeführt, die in den genannten Sprachen eine gleiche Datendarstellung besitzen und daher problemlos als Parameter übergeben werden können, soweit dies für den Aufruf von Window Services erforderlich ist.

Datentypen aus Programmiersprachen, die als Parameter für DIV übergeben werden können:

Compiler	**Datentyp**	
	Binär Wort	**String**
PL/1	BIN FIXED (31)	CHAR (i)
COBOL	PIC S9(i) COMP (i = 5,...,9)	USAGE DISPLAY
C	long	char <var>
FORTRAN	INT * 4	(<size>) CHAR * i

Der Adapter Window Services wird ausgeliefert als Laufzeit-Bibliothek (OML) in einer Datei mit dem Namen SYSLIB.DWS.110 unter der Kennung TSOS des Systemverwalters.

Bei Aufruf von DIV über Window Services können einige DIV-Funktionen nicht oder nur eingeschränkt genützt werden:

– Bei Window Services können Fenster nicht in einem Datenraum liegen (SPID entfällt).
– Die Funktion LOCVIEW=MAP gibt es nicht bei Window Services.
– RELEASE=YES (Funktion RESET) wird von Window Services nicht unterstützt.
– Statt der Möglichkeit, bei seq. Bearbeitung die Zahl der im Voraus einzulesenden Seiten anzugeben (PFCOUNT), wird bei Window Services nur der sequentielle Zugriff angezeigt (USAGE).
  USAGE='RANDOM' wirkt wie PFCOUNT=0,
  USAGE='SEQ' wird auf PFCOUNT=15 abgebildet.
– Als Returncode wird nur der Main-Returncode zur Verfügung gestellt.

CALL-Aufrufe

Beim CALL-Aufruf der entsprechenden Funktionen müssen verschiedene Parameter übergeben werden. Die folgende Tabelle zeigt eine Zusammenfassung aller möglichen Werte.

Feldname	Datentyp PL/1-Umgebung	Kurzerklärung
DISPOS	CHAR(6);	/* bei Funktion=MAP: 'OBJECT' 'UNCHNG'
		/* bei Funktion UNMAP: 'UNCHNG' 'FRESH'
DMS_CODE	BIN FIXED (31);	
FILENAME	CHAR(54);	
ID	CHAR (8);	/* Identifikation des OPEN,
		/* (Ausgabe-Parameter)
LINKNAME	CHAR(8);	
OFFSET	BIN FIXED (31);	
OPEN-MODE	CHAR(5);	/* 'INPUT' INOUT' 'OUTIN'
RETURNCODE	BIN FIXED (31);	
SHARUPD-MODE	CHAR(4);	/* 'NO' 'WEAK' 'YES'
SIZE	BIN FIXED (31);	/* Dateigröße,
		/* (Ausgabe-Param.)
SPAN	BIN FIXED (31);	
USAGE	CHAR(6);	/* ' RANDOM' ' SEQ'
WINDOW	AREA(1) BASED ...;	/* Ausrichtung auf
		/* Seitengrenze !!

Im Folgenden werden die CALL-Aufrufe für die einzelnen DIV-Funktionen (in PL/1) aufgeführt. Welche Parameter bei dem jeweilgen CALL-Aufruf angegeben werden müssen, ist den einzelnen CALL-Aufrufen zu entnehmen.

DIV-Funktion OPEN

```
CALL DWSOPEN  (LINKNAME, FILENAME, OPEN-MODE, SHARUPD-MODE, ID,
               SIZE, RETURNCODE, DMS-CODE);
```

DIV-Funktion MAP

```
CALL DWSMAP   (ID, OFFSET, SPAN, WINDOW, USAGE, DISPOS,
               RETURNCODE);
```

DIV-Funktion SAVE

```
CALL DWSSAVE  (ID, OFFSET, SPAN, SIZE, RETURNCODE);
```

Funktion REFRESH (entspricht der DIV-Funktion "RESET")

```
CALL DWSREFR  (ID, OFFSET, SPAN, RETURNCODE);
```

DIV-Funktion UNMAP

```
CALL DWSUNMP  (ID, OFFSET, SPAN, WINDOW, DISPOS, RETURNCODE);
```

DIV-Funktion CLOSE

```
CALL DWSCLS   (ID, RETURNCODE, DMS_CODE);
```

5.4 Kennsatzformate

5.4.1 Familie der Bandanfangs-Kennsätze

Jeder Datenträger enthält mindestens einen Bandanfangs-Kennsatz (VOL1) und höchstens neun. Die Bandanfangs-Kennsätze VOL2 bis VOL9 sind optional.

Erster Bandanfangs-Kennsatz (VOL1)

Der erste Bandanfangs-Kennsatz identifiziert den Datenträger, den Eigentümer, die Zugriffsbedingungen, die Implementation und die Ausgabenummer der verwendeten Norm.

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	VOL
4	Kennsatznummer	1	1
5 bis 10	Bandkennzeichen	6	„a“-Zeichen. Vom Eigentümer fest zugeordnet, um das Band zu kennzeichnen
11	Datenträgerzugriffsvermerk	1	„a“-Zeichen. Zeigt Einschränkungen bezüglich des Zugriffs zu den Daten auf diesem Band an. Ein Leerzeichen bedeutet, dass keine Einschränkung für den Zugriff auf den Datenträger gegeben ist. Ein beliebiges anderes „a“-Zeichen bedeutet, dass es besondere Einschränkungen für den Zugriff auf den Datenträger gibt. BS2000: Leerzeichen oder '0' : unbeschränkter Zugriff '1' : Zugriff nur vom Eigentümer möglich
12 bis 24	Reserviert für spätere Normung	13	Leerzeichen
25 bis 37	System-Code	13	„a“-Zeichen. Identifiziert die Implementation, die die Bandanfangs-Kennsätze aufgezeichnet hat.
38 bis 51 38 bis 41 42 bis 49 50 bis 51	Eigentümerkennzeichen	14 4 8 2	„a“-Zeichen. Identifiziert den Eigentümer des Bandes BS2000: Leerzeichen „a“-Zeichen: Benutzerkennung Leerzeichen

Stelle	Feldname	Länge	Feldinhalt
52 bis 79	Reserviert für spätere Normung	28	Leerzeichen
80	Normvermerk	1	Zeigt die Ausgabe der Norm an, der die Kennsätze und Datenformate auf diesem Band entsprechen. 4 bedeutet: DIN 66029-4 (Ausgabe September 1987) 3 bedeutet: DIN 66029-3 (Ausgabe Mai 1979) 2 bedeutet: DIN 66029-2 (Ausgabe Juni 1976) 1 bedeutet: DIN 66029-1 (Ausgabe August 1972)

Weitere Bandanfangs-Kennsätze (VOL2 bis VOL9)

Die weiteren Bandanfangs-Kennsätze sind optional. Die Implementationen von Fujitsu erzeugen sie nicht.

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	VOL
4	Kennsatznummer	1	Ziffer 2 bis 9
5 bis 80	Reserviert für die Implementation	76	Für Aufzeichnung und inhaltliche Bedeutung dieses Feldes gibt es weder Festlegungen noch Einschränkungen

5.4.2 Familie der Benutzer-Bandanfangs-Kennsätze (UVL1 bis UVL9)

Die Benutzer-Bandanfangs-Kennsätze sind optional.

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	UVL
4	Kennsatznummer	1	Ziffer 1 bis 9
5 bis 80	Reserviert für den Betreiber	76	Für Aufzeichnung und inhaltliche Bedeutung dieses Feldes gibt es weder Festlegungen noch Einschränkungen

BS2000 liefert die UVL-Kennsätze an den Benutzer.

5.4.3 Familie der Dateianfangs-Kennsätze (HDR1 bis HDR9)

Jede Datei oder jeder Dateiabschnitt enthält mindestens zwei Dateianfangs-Kennsätze (HDR1 und HDR2) und höchstens neun. Die Dateianfangs-Kennsätze HDR3 bis HDR9 sind optional.

Erster Dateianfangs-Kennsatz (HDR1)

Der erste Dateianfangs-Kennsatz identifiziert einen Dateiabschnitt, beschreibt seine Lage innerhalb der Dateimenge und bestimmt gewisse Merkmale des Dateiabschnitts.

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	HDR
4	Kennsatznummer	1	1
5 bis 21	Dateiname	17	„a“-Zeichen. Kennzeichnet die Datei
22 bis 27	Dateimengenkenn-zeichen	6	„a“-Zeichen. Kennzeichnet die Dateimenge, zu der diese Datei gehört
28 bis 31	Dateiabschnitts-nummer	4	„n“-Zeichen. Kennzeichnet den Dateiabschnitt Die Nummer des ersten Dateiabschnitts einer Datei ist 0001. Diese Nummer wird für jeden folgenden Datei-abschnitt dieser Datei um eins erhöht.
32 bis 35	Dateifolgenummer	4	„n“-Zeichen. Kennzeichnet die Datei der Dateimenge. Die Dateifolgenummer der ersten Datei in einer Dateimen-ge ist 0001. Diese Nummer wird bei jeder folgenden Datei einer Dateimenge um eins erhöht. In allen Kennsätzen einer bestimmten Datei muss dieses Feld die gleiche Zahl enthalten, unabhängig davon, ob die Datei auf einem oder mehreren Bändern liegt.
36 bis 39	Generationsnum-mer	4	„n“-Zeichen. Unterscheidet die aufeinander folgenden Fortschreibungen der Datei von 0001 bis 9999.
40 bis 41	Versionsnummer	2	„n“-Zeichen. Unterscheidet die aufeinander folgenden Wiederholungen einer Generation
42 bis 47	Erstellungsdatum	6	Leerzeichen oder „n“-Zeichen. Gibt das Datum der Erstel-lung des Dateiabschnitts an. Ein Leerzeichen für das 20. Jahrhundert; die Ziffer 0 für das 21. Jahrhundert, gefolgt von zwei „n“-Zeichen für das Jahr (00 bis 99) innerhalb des Jahrhunderts, gefolgt von drei „n“-Zeichen für den Tag des Jahres (001 bis 366). Der Wert 00000 in den letzten fünf Stellen zeigt an, dass das Erstellungsdatum ohne Bedeutung ist.

Stelle	Feldname	Länge	Feldinhalt
48 bis 53	Verfallsdatum	6	Leerzeichen oder „n“-Zeichen. Gibt das früheste Datum an, ab dem der Dateiabschnitt gelöscht werden darf. Format wie Feld Erstellungsdatum (Stelle 42 bis 47). Der Wert 00000 in den letzten fünf Stellen zeigt an, dass das Verfallsdatum ohne Bedeutung ist und der Dateiabschnitt veraltet ist.
54	Dateizugriffsvermerk	1	„a“-Zeichen. Zeigt Einschränkungen bezüglich des Zugriffs zu den Daten dieser Datei an. Ein Leerzeichen bedeutet, dass keine Einschränkung für den Zugriff auf die Datei gegeben ist. Ein beliebiges anderes „a“-Zeichen bedeutet, dass es besondere Einschränkungen für den Zugriff auf die Datei gibt. BS2000: '1' oder '3' : Band- oder Dateieigentümer haben Zugriff
55 bis 60	Blockzähler	6	000000
61 bis 73 61 bis 65 66 bis 73	System-Code	13 5 8	„a“-Zeichen. Identifiziert die Implementation, durch die die Kennsätze erzeugt werden. BS2000: BS2000 Leerzeichen
74 bis 80	Reserviert für spätere Normung	7	Leerzeichen

Zweiter Dateianfangs-Kennsatz (HDR2)

Der zweite Dateianfangs-Kennsatz beschreibt Merkmale der Datei und der Implementation.

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	HDR
4	Kennsatznummer	1	2
5	Satzformat	1 1	F, D oder S und lt. DIN nicht unterstützt V und U. Gibt das Format der Sätze der Datei an. – F: alle Sätze der Datei haben eine feste Satzlänge. – D: alle Sätze haben eine variable Länge und die Anzahl der Zeichen als Dezimalzahl ist im Satz selbst angegeben. – S: alle Sätze sind segmentiert. – V: alle Sätze haben eine undefinierte Länge (lt. DIN nicht unterstützt). – U: alle Sätze haben eine variable Länge und die Anzahl der Zeichen als Dualzahl ist im Satz selbst angegeben (lt. DIN nicht unterstützt).
6 bis 10 6 bis 7 8 bis 10	Blocklänge STD-Blöcke	5	„n“-Zeichen. Gibt die maximale Anzahl der Zeichen je Block der Datei an BS2000: Bei Normvermerk 1 kann der Inhalt wie folgt sein: '80': Kennzeichen für STD-Block „n“-Zeichen. Gibt die Anzahl der STD-Blöcke an.
11 bis 15	Satzlänge	5	„n“-Zeichen. Kennzeichnet die Satzlänge in Verbindung mit dem Satzformat (Stelle 5) Bei Satzformat F enthält dieses Feld die tatsächliche Satzlänge. Bei Satzformat D und V enthält dieses Feld die maximale Länge einschließlich Satzlängenwort (SLW). Bei Satzformat S enthält dieses Feld die maximale Satzlänge, wobei die Segmentkontrollwörter (SKW) ausgenommen sind. Der Inhalt 00000 in diesem Feld bedeutet beim Satzformat S, dass die Satzlänge größer als 99999 sein kann. Bei Satzformat U enthält dieses Feld die maximale Anzahl der Zeichen, die ein Satz enthalten kann.

Stelle	Feldname	Länge	Feldinhalt
16 bis 50	Reserviert für die Implementation	35	Für Aufzeichnung und inhaltliche Bedeutung dieses Feldes gibt es weder Festlegungen noch Einschränkungen.
16	Schreibdichte	1	Belegung von BS2000 bis zur Unterstützung der DIN Stufe 4: 0 200 Bpi 1 556 Bpi 2 800 Bpi 3 1600 Bpi 4 6250 Bpi
17	Datenposition	1	Anzeige bei Spulenwechsel 0 nein 1 ja
18 bis 34	Auftragsschrift-Kennung	17	Durch Prozessverwalter zugewiesene Kennung
35 bis 36	Schreibdichte bei Magnetbandkassetten	2	' ␣␣' nicht komprimiert ' P␣' komprimiert
47 bis 50	Dateinamencode	4	Wird nur bis DIN 66029-1 verwendet, wenn die Stellen 6 bis 7 STD-Blöcke enthalten.
51 bis 52	Pufferverschiebung	2	„n“-Zeichen. Gibt die Länge (in Zeichen) eines zusätzlichen Feldes an, das am Anfang eines jeden Datenblocks eingefügt ist
53 bis 80	Reserviert für spätere Normung	28	Leerzeichen

Dritter Dateianfangs-Kennsatz (HDR3)

Der HDR3-Kennsatz enthält für den Dateieigentümer den vollständigen Dateinamen, die Kennwörter und die Zugriffsart.

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	HDR
4	Kennsatznummer	1	3
5 bis 12	Eigentümerkenn-zeichen	8	Identifiziert den Eigentümer der Datei (Benutzerkennung).
13 bis 56	Dateiname	44	Die ersten 44 Zeichen des Namens der Datei bzw. der Dateigeneration, zu der die Datei gehört.
57 bis 60	Lesekennwort	4	Bezeichnet ein zum Lesen der Datei erforderliches Kennwort
61 bis 64	Schreibkennwort	4	Bezeichnet ein zum Lesen und Schreiben der Datei erforderliches Kennwort
65 bis 68	Ablaufkennwort	4	Bezeichnet ein, um einen in der Datei befindlichen Lademodul ablaufen zu lassen, erforderliches Kennwort
69	Zugriffsart	1	Gibt die zulässige Zugriffsart an: 0: Lese- und Schreibzugriff erlaubt 1: nur Lesezugriff erlaubt
70 bis 80	Reserviert	11	Leerzeichen

Weitere Dateianfangs-Kennsätze (HDR4 bis HDR9)

Die weiteren Dateianfangs-Kennsätze enthalten implementationsabhängige Informationen.

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	HDR
4	Kennsatznummer	1	Ziffer 4 bis 9
5 bis 80	Reserviert für die Implementation	76	Für Aufzeichnung und inhaltliche Bedeutung dieses Feldes gibt es weder Festlegungen noch Einschränkungen

5.4.4 Familie der Benutzer-Dateianfangs-Kennsätze (UHL)

Die Benutzer-Dateianfangs-Kennsätze sind optional.

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	UHL
4	Kennsatznummer	1	„a“-Zeichen. Vom Benutzer festzulegen BS2000: „b“-Zeichen
5 bis 80	Reserviert für den Benutzer	76	Für Aufzeichnung und inhaltliche Bedeutung dieses Feldes gibt es weder Festlegungen noch Einschränkungen

BS2000 unterstützt bis zu 255 UHL-Kennsätze. Die Kennsatznummer (Stelle 4) wird vom Benutzer festgelegt.

5.4.5 Familie der Bandende-Kennsätze (EOV1 bis EOV9)

Jeder Dateiabschnitt, der auf einer Folgespule fortgesetzt wird, enthält mindestens zwei Bandende-Kennsätze (EOV1 und EOV2) und höchstens neun. Die Bandende-Kennsätze EOV3 bis EOV9 sind optional.

Erster Bandende-Kennsatz (EOV1)

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	EOV
4	Kennsatznummer	1	1
5 bis 54	gleich den entsprechenden Feldern in HDR1	50	gleich den entsprechenden Feldern in HDR1
55 bis 60	Blockzähler	6	„n“-Zeichen. Gibt die Anzahl der Datenblöcke an, die den Dateiabschnitt bilden
61 bis 80	gleich den entsprechenden Feldern in HDR1	20	gleich den entsprechenden Feldern in HDR1

Zweiter Bandende-Kennsatz (EOV2)

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	EOV
4	Kennsatznummer	1	2
5 bis 80	gleich den entsprechenden Feldern in HDR2	76	gleich den entsprechenden Feldern in HDR2

Dritter Bandende-Kennsatz (EOV3)

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	EOV
4	Kennsatznummer	1	3
5 bis 80	gleich den entsprechenden Feldern HDR3	76	gleich den entsprechenden Feldern in HDR3

Weitere Bandende-Kennsätze (EOV4 bis EOV9)

Die Bandende-Kennsätze enthalten implementationsabhängige Informationen.

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	EOV
4	Kennsatznummer	1	4 bis 9
5 bis 80	Reserviert für die Implementation	76	Für Aufzeichnung und inhaltliche Bedeutung dieses Feldes gibt es weder Festlegungen noch Einschränkungen

5.4.6 Familie der Dateiende-Kennsätze (EOF1 bis EOF9)

Jede Datei enthält mindestens zwei Dateiende-Kennsätze (EOF1 und EOF2) und höchstens neun. Die Dateiende-Kennsätze EOF3 bis EOF9 sind optional.

Erster Dateiende-Kennsatz (EOF1)

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	EOF
4	Kennsatznummer	1	1
5 bis 54	gleich den entsprechenden Feldern in HDR1	50	gleich den entsprechenden Feldern in HDR1
55 bis 60	Blockzähler	6	„n“-Zeichen. Gibt die Anzahl der Datenblöcke an, die den Dateiabschnitt bilden
61 bis 80	gleich den entsprechenden Feldern in HDR1	20	gleich den entsprechenden Feldern in HDR1

Zweiter Dateiende-Kennsatz (EOF2)

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	EOF
4	Kennsatznummer	1	2
5 bis 80	gleich den ensprechenden Feldern in HDR2	76	gleich den entsprechenden Feldern in HDR2

Dritter Dateiende-Kennsatz (EOF3)

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	EOF
4	Kennsatznummer	1	3
5 bis 80	gleich den entsprechenden Feldern HDR3	76	gleich den entsprechenden Feldern in HDR3

Weitere Dateiende-Kennsätze (EOF4 bis EOF9)

Die weiteren Dateiende-Kennsätze enthalten implementationsabhängige Informationen.

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	EOF
4	Kennsatznummer	1	4 bis 9
5 bis 80	Reserviert für die Implementation	76	Für Aufzeichnung und inhaltliche Bedeutung dieses Feldes gibt es weder Festlegungen noch Einschränkungen

5.4.7 Familie der Benutzer-Dateiende-Kennsätze (UTL)

Die Benutzer-Dateiende-Kennsätze sind optional.

Aufbau

Stelle	Feldname	Länge	Feldinhalt
1 bis 3	Kennsatzname	3	UTL
4	Kennsatznummer	1	„a“-Zeichen. Vom Benutzer festzulegen BS2000: „b“-Zeichen
5 bis 80	Reserviert für den Benutzer	76	Für Aufzeichnung und inhaltliche Bedeutung dieses Feldes gibt es weder Festlegungen noch Einschränkungen

BS2000 unterstützt bis 255 UTL-Kennsätze. Die Kennsatznummern (Stelle 4) werden vom Benutzer festgelegt.

5.4.8 Verarbeitung der Kennsatzfelder

Anforderungen an ein sendendes System

Dateien

Der Implementation sind von dem Anwendungsprogramm die Datensätze der Dateien zu übergeben.

Kennsätze

Der Benutzer muss die in jedem der nachfolgend aufgeführten Kennsatzfelder benötigte Information zur Aufzeichnung an die Implementation übergeben, andernfalls muss die Implementation diese Information liefern.

Für jeden Datenträger einer Datenträgermenge:

– Bandkennzeichen (VOL1, Stelle 5 bis 10)
– Datenträgerzugriffsvermerk (VOL1, Stelle 11)

Für jede Datei einer Dateimenge:

– Dateizugriffsvermerk (HDR1, Stelle 54)

Wenn die Implementation dem Benutzer erlaubt, die Information bereitzustellen, die in jedem der nachfolgend aufgeführten Kennsatzfelder zu verarbeiten ist, dann muss die Implementation diese Information verarbeiten. Wenn der Benutzer diese Information nicht liefert, muss die Implementation die Information bereitstellen.

Für jeden Datenträger einer Datenträgermenge:

– Eigentümerkennzeichen (VOL1, Stelle 38 bis 51)

Für jede Datei einer Dateimenge:

– Dateimengenkennzeichen (HDR1, Stelle 22 bis 27)

Die Implementation muss dem Anwendungsprogramm erlauben, die Information bereitzustellen, die in jedem der nachfolgend aufgeführten Kennsatzfelder zu verarbeiten ist. Wenn das Anwendungsprogramm diese Information nicht liefert, muss die Implementation die Information für das jeweilige Feld bereitstellen.

Für jede Datei einer Dateimenge:

– Dateiname (HDR1, Stelle 5 bis 21)
– Satzformat (HDR2, Stelle 5)
– Blocklänge (HDR2, Stelle 6 bis 10)
– Satzlänge (HDR2, Stelle 11 bis 15)

Wenn die Implementation dem Anwendungsprogramm erlaubt, die Information bereitzustellen, die in jedem der nachfolgend aufgeführten Kennsatzfelder zu verarbeiten ist, dann muss die Implementation diese Information verarbeiten. Wenn das Anwendungsprogramm diese Information nicht liefert, muss die Implementation die Information bereitstellen.

Für jede Datei einer Dateimenge:

– Generationsnummer (HDR1, Stelle 36 bis 39)
– Versionsnummer (HDR1, Stelle 40 und 41)

Für jeden Dateiabschnitt einer Dateimenge:

– Erstellungsdatum (HDR1, Stelle 42 bis 47)
– Verfallsdatum (HDR1, Stelle 48 bis 53)

Wenn die Implementation in der Lage ist, eine Familie von Benutzer-Bandanfangs-Kennsätzen (UVL) zu verarbeiten, dann muss die Implementation dem Benutzer erlauben, die zu verarbeitende Information aus den nachfolgend aufgeführten Kennsatzfeldern bereitzustellen. Die Verarbeitung der entsprechenden Kennsätze wird nicht gefordert, wenn der Benutzer die Information nicht bereitstellt.

Für jeden Kennsatz aus einer Familie von Benutzer-Bandanfangs-Kennsätze, die auf jedem Band einer Bandmenge aufgezeichnet sind, gilt:

– Reserviert für den Benutzer (UVL, Stelle 5 bis 80)

Wenn die Implementation in der Lage ist, eine Familie von Benutzer-Dateianfangs-Kennsätzen (UHL) oder von Benutzer-Dateiende-Kennsätzen (UTL) zu verarbeiten, dann muss die Implementation dem Anwendungsprogramm erlauben, die Information bereitzustellen, die in den nachfolgend aufgeführten Kennsatzfeldern für eine Kennsatzfamilie einzutragen ist. Die Verarbeitung der entsprechenden Kennsätze wird nicht gefordert, wenn das Anwendungsprogramm die Informationen nicht bereitstellt.

Für jeden Kennsatz in einer Familie von Benutzer-Dateianfangs- und Benutzer-Dateiende-Kennsätze eines Magnetbandes gilt:

– Kennsatznummer (UHL/UTL, Stelle 4)
– Reserviert für den Benutzer (UHL/UTL, Stelle 5 bis 80)

Die Implementation kann dem Benutzer bezüglich der Satzlänge (HDR2, Stelle 11 bis 15) die nachfolgend beschriebenen Beschränkungen auferlegen.

Wenn die Implementation segmentierte Sätze verarbeitet, kann sie eine maximale Satzlänge festlegen. Diese Grenze soll nicht unter der maximalen zulässigen Blocklänge liegen, abzüglich der Länge des Pufferverschiebungsfeldes und abzüglich der Länge des Segmentkontrollwortes (SKW).

Wenn die Implementation Sätze variabler Länge verarbeitet, kann sie eine maximale Satzlänge festlegen, die der maximalen Blocklänge, abzüglich der Länge des Pufferverschiebungsfeldes und abzüglich der Länge des Satzlängenwortes (SLW) entspricht.

Anforderungen an ein empfangendes System

Dateien

Die Implementation muss für das Anwendungsprogramm den Inhalt der Datensätze und die Länge jedes Datensatzes bereitstellen. Das Segmentkontrollwort (SKW) und das Satzlängenwort (SLW) sind nicht Bestandteile des Datensatzes.

Kennsätze

Die Implementation muss dem Benutzer gestatten, ausreichende Information bereitzustellen, um die von ihm angeforderten Dateien sowie den Datenträger auszuwählen, auf dem diese Dateien aufgezeichnet sind.

Die Implementation muss für den Betreiber die Information bereitstellen, die in den nachfolgend aufgeführten Kennsatzfeldern enthalten ist.

Für jeden Datenträger einer Datenträgermenge:

– Bandkennzeichen (VOL1, Stelle 5 bis 10)
– Datenträgerzugriffsvermerk (VOL1, Stelle 11)

Für jede Datei einer Dateimenge:

– Dateizugriffsvermerk (HDR1, Stelle 54)

Die Implementation muss für das Anwendungsprogramm die Information bereitstellen, die in den nachfolgend aufgeführten Kennsatzfeldern enthalten ist.

Für jede Datei einer Dateimenge:

– Dateiname (HDR1, Stelle 5 bis 21)
– Satzformat (HDR2, Stelle 5)
– Blocklänge (HDR2, Stelle 6 bis 10)
– Satzlänge (HDR2, Stelle 11 bis 15)

Die Implementation braucht für den Benutzer nicht die Information bereitzustellen, die in den nachfolgenden Kennsatzfeldern enthalten ist.

Für jeden Datenträger einer Datenträgermenge:

– Eigentümerkennzeichen (VOL1, Stelle 38 bis 51)

Für jede Datei einer Dateimenge:

– Dateimengenkennzeichen (HDR1, Stelle 22 bis 27)
– Generationsnummer (HDR1, Stelle 36 bis 39)
– Versionsnummer (HDR1, Stelle 40 und 41)

Für jeden Dateiabschnitt einer Dateimenge:

– Erstellungsdatum (HDR1, Stelle 42 bis 47)
– Verfallsdatum (HDR1, Stelle 48 bis 53)

Wenn die Implementation in der Lage ist, dem Benutzer die Information zur Verfügung zu stellen, die in der Familie der Benutzer-Bandanfangs-Kennsätze (UVL) aufgezeichnet ist, dann muss die Information bereitgestellt werden, die in den nachfolgenden Kennsatzfeldern enthalten ist.

Für jeden Kennsatz einer Familie von Benutzer-Bandanfangs-Kennsätzen:

– Reserviert für den Benutzer (UVL, Stelle 5 bis 80)

Wenn die Implementation in der Lage ist, dem Benutzer die Information zur Verfügung zu stellen, die in der Familie von Benutzer-Dateianfangs-Kennsätzen (UHL) oder von Benutzer-Dateiende-Kennsätzen (UTL) aufgezeichnet ist, dann muss die Information bereitgestellt werden, die in den nachfolgenden Kennsatzfeldern enthalten ist.

– Kennsatznummer (UHL/UTL, Stelle 4)
– Reserviert für den Benutzer (UHL/UTL, Stelle 5 bis 80)

5.5 DVS-Pseudo-Programmabschnitte (DSECTS)

Neben dem MF-Operanden stehen für einige Operandenlisten sowie für Pseudo-Programmabschnitte von DVS-Tabellen, FCB- und Katalogeinträgen den Anwenderprogrammen DSECTs zur Verfügung. Namen und Größen der einzelnen Felder sind definiert; die Anordnung der Abschnitte kann jedoch Änderungen unterworfen sein.

Der Anwender, der an der Auflösung eines bestimmten Pseudoprogrammabschnittes interessiert ist, kann einen Makroaufruf in der folgenden Form verwenden:

Operation	Operanden
makroname	[D][,{prefix / *}][,PARMOD={24 / 31}]

Für „makroname“ siehe Tabelle auf der folgenden Seite

Voreinstellung es wird keine DSECT generiert, Präfix ist der Buchstabe I

D
Gibt an, dass der Makroaufruf verwendet wird, um eine DSECT zu generieren.

prefix
Präfix (1 Zeichen), das allen Namen in der DSECT vorangestellt werden soll.

*****
Gibt an, dass kein Präfix verwendet werden soll.

PARMOD
Der PARMOD-Operand kann bei den DSECT-Makros IDFCB, IDMCB, IDFST, IDPPL und DMARD verwendet werden. Er gibt an, welche Operandenliste der DSECT generiert werden soll.

Voreinstellung: der durch den Makro GPARMOD oder durch den Assembler voreingestellte Wert für den Generierungsmodus

= 24
Es wird die 24-Bit-adressierungsmodus-abhängige Operandenliste generiert.

= 31
Es wird die adressierungsmodus-unabhängige Operandenliste generiert; ihre Namen können von denen der PARMOD-24-Operandenliste abweichen.

Hinweise zur Programmierung

– Der Generierungsmodus kann für alle Makroaufrufe eines Programms mit dem Makro GPARMOD global eingestellt werden. Der PARMOD-Operand in den DVS-Makroaufrufen setzt die Voreinstellung durch einen GPARMOD-Aufruf oder (bei fehlendem GPARMOD) die Voreinstellung des Assemblers außer Kraft.

– Alle PARMOD-Angaben für eine Operandenliste (z.B. MF=E/L/D und die entsprechenden DSECT-Makros) und für eine Datei müssen den gleichen Wert haben.

Pseudo-Programmabschnitte (DSECTs) des DVS für den Anwender:

Makro-Name	DSECT-Beschreibung
IDBPL	BTAM-Operandenliste
IDCAT	CATALOG (CATAL) Operandenliste (altes Format)
IDCE	Katalogeintrag
IDCEG	Katalogeintrag, Zusatz für Dateigenerationsgruppen
IDCEX	Katalogeintrag Erweiterung
IDCHA	CHANGE (CHNGE) Operandenliste
IDCOP	COPY-Makroaufruf Operandenliste
IDECB	UPAM-Ereignissteuerblock
IDEE	Katalogeintrag, Extent-Liste
IDEMS	Fehlermeldungen des DVS (siehe Seite 875)
IDERS	ERASE-Makroaufruf Operandenliste (für VERSION=0)
IDFCB	FCB (TU-Abschnitt; gültig für alle FCB-Formate)
IDFCBE	FCB-Erweiterung (für 24-bit-TU-FCB)
IDFST	FSTAT-Makroaufruf Operandenliste
DMAIMP	IMPORT-Makroaufruf Operandenliste (altes Format)
IDMCB	EAM-Steuerblock
IDOST	Information über eröffnete Dateien (siehe OSTAT-Makro, Seite 755)
IDPFL	FILE-Makroaufruf Operandenliste (altes Format)
IDPFX	FILE-Makroaufruf Operandenlisten-Erweiterung (altes Format)
IDPPL	UPAM-Operandenliste
IDREL	REL-Makroaufruf (RELEASE) Operandenliste
IDVT	Datenträger-Kennsatz-Eintrag (VT=Volume Table)
IDVRF	VERIF-Makroaufruf Operandenliste
DMADR	RDTFT-Ausgabeformat (bei Angabe des LINK-Operanden) (altes Format)
DMARD	RDTFT-Makroaufruf Operandenliste (altes Format)

5.6 Makroformate ersetzter Makros

Für die nachfolgenden Makros wurden neue Makros eingeführt. Die alten Makros werden aus Kompatibilitätsgründen weiter unterstützt, ihre Funktionalität jedoch nicht mehr erweitert. Hier werden die Formate dieser Makros gezeigt. Die Operandenbeschreibungen entsprechen denen der neuen Makros, siehe vorne.

COPY – Datei kopieren

Der COPY-Makro wurde durch den Makro COPFILE ersetzt. Die Beschreibung des neuen COPFILE-Makros und die Beschreibung der entsprechenden COPY-Operanden befindet sich auf Seite 218ff.

Operation	Operanden
COPY	pfadname$_1$,pfadname$_2$ [,SAME][,WRITE={REPLACE / NEW}][,BLKCTRL={IGNORE / CHECK}] [,IGNORE={SOURCE / TARGET / (SOURCE,TARGET)}]

REL – TFT-Eintrag löschen

Der REL-Makro wurde durch den Makro RELTFT ersetzt. Die Beschreibung des neuen RELTFT-Makros und die Beschreibung der entsprechenden REL-Operanden befindet sich auf Seite 785ff.

Operation	Operanden
REL	MF=(E,{adr / (r)})
	name[,KEEP][,UNLOAD][,MF=L]

Fachwörter

Im Folgenden werden kurze Definitionen einiger in der vorliegenden Beschreibung verwendeter Fachbegriffe gegeben.

Abschnittsmarke (Tape Mark)

Ein Bandblock, der die Grenze sowohl zwischen Datenblöcken und Kennsatzgruppen als auch zwischen bestimmten Kennsatzgruppen anzeigt. Der Aufbau der Abschnittsmarke ist in den entsprechenden Normen über Magnetbänder angegeben.

ACL (Access Control List)

Der Dateischutz mit einer ACL (Zugriffskontroll-Liste) wird bereits seit SECOS V4.0 nicht mehr unterstützt. Das Dateimerkmal ACL ist noch im Katalogeintrag enthalten, enthält aber im Normalfall den Wert NO (kein ACL-Schutz).

Sollte der Fall auftreten, dass eine Datei ACL-geschützt ist (evtl. möglich nach Restaurierung aus einem Langzeitarchiv), ist der Dateizugriff nicht möglich. In diesem Fall kann nur die Systembetreuung (Privileg TSOS) den Zugriff wieder ermöglichen (z.B. mit /COPY-FILE und IGNORE-PROTECTION eine zugreifbare Kopie anlegen oder mit dem Makro CATAL den ACL-Indikator im Katalogeintrag zurücksetzen).

Alias Catalog Service (ACS)

Dateien und Jobvariablen können über Aliasnamen angesprochen werden und zusammen mit der Zuordnung zur realen Datei/JV in speziellen Katalogen, den Aliaskatalogen, hinterlegt werden. Der Alias Catalog Service (ACS) umfasst drei Grundfunktionen: Aliasnamen-Vereinbarung, Catid-Einfügung für temporäre Spooldateien und Präfix-Einfügung.

alphanumerisch

alphanumerische Zeichen umfassen *alpha*betische und *numerische* Zeichen, d.h. die Buchstaben A-Z und die Ziffern 0-9.

Archivnummer (= VSN, Volume Serial Number)
Sie besteht aus 6 Zeichen und wird dem Datenträger bei der Initialisierung (VOLIN bzw. INIT) zugeteilt. Sie ist im Standard-Datenträgerkennsatz enthalten und dient der Identifizierung des Datenträgers.

Auftragsnummer (= TSN, Task Sequence Number)
die vom System für den Auftrag vergebene (laufende) Nummer, mit der der Anwender bei einigen Kommandos einen Auftrag identifizieren kann.

Auftrag
Gesamtheit aller Abläufe zwischen den Benutzerkommandos SET-LOGON-PARAMETERS bzw. LOGON und EXIT-JOB bzw. LOGOFF. Dabei ist es unwichtig, ob der Auftrag bereits vollständig definiert ist (Batchbetrieb) oder ob die einzelnen Schritte erst beim Ablauf festgelegt werden (Dialogbetrieb).

Band (Volume)
Eine auswechselbare Einheit des Datenträgers Magnetband. Ein Band kann eine Datei ganz oder teilweise enthalten. Es kann auch mehrere Dateien und/oder einen bzw. mehrere Datei-Abschnitte enthalten.

Bandmenge (Volume Set)
Das Band oder die Bänder, auf denen die Dateien einer Dateimenge aufgezeichnet sind.

Batchbetrieb
Auftrag, der mit dem ENTER-JOB- bzw. ENTER-PROCEDURE-Kommando gestartet wurde; im Gegensatz zum Dialogbetrieb ist der Ablauf vordefiniert und in einer ENTER-Datei (Start mit ENTER-JOB) bzw. in einer Prozedurdatei (Start mit ENTER-PROCEDURE) festgelegt.

Batchverarbeitung
Siehe Batchbetrieb

Block
Siehe PAM-Seite, Datenblock

BS2000-Datei
Bezeichnet eine Datei, die ausschließlich von BS2000 angelegt und bearbeitet wird. BS2000-Dateien auf Net-Storage (FILE-TYPE=BS2000) werden seit BS2000/OSD-BC V9.0 bedient. Sie liegen direkt auf einem Net-Storage-Volume. Offene Systeme dürfen ausschließlich lesend darauf zugreifen.

CALL-Prozedur
: Kommando-/Anweisungsfolge, die in einem Auftrag abläuft; Aufruf mit CALL-PROCEDURE-Kommando (siehe Handbuch „Kommandos“ [3]).

Catid (= Katalogkennung)
: Kennzeichen eines Pubsets (Siehe Pubset-Id); wird im vollständigen Dateinamen/Pfadnamen in der Form :catid: angegeben.

Datei
: Sätze, die zueinander in Beziehung stehen, werden in einer benannten Einheit, der Datei, zusammengefasst. Dateien sind beispielsweise: konventionelle Ein-/Ausgabedaten von Programmen; Lademodule und Bindemodulbibliotheken; Textinformation, die mit dem Dateiaufbereiter erstellt und verarbeitet wird.

Dateiabschnitt (File Section)
: Der Teil einer Datei, der auf einem einzigen Band aufgezeichnet ist.

Dateikettungsname
: maximal 8 Zeichen langer Name, der die Verbindung zwischen dem FCB-Makroaufruf oder Dateisteuerblock und der Datei über Task File Table herstellt.

Dateimenge (File Set)
: Eine Menge von Dateien, die aufeinander folgend auf einem oder mehreren Bändern aufgezeichnet ist. Zwischen den Abschnitten einer Datei innerhalb einer Dateimenge dürfen sich keine Abschnitte anderer Dateien befinden.

Datenblock
: Ein Block, der einen oder mehrere Sätze einer Datei enthält.

Dialogbetrieb
: Ein Auftrag, der von einem entfernten Datenplatz eingeleitet wird und abläuft; der Ablauf ist nicht im Voraus festgelegt.

Doppel-Abschnittsmarke (Double Tape Mark)

Zwei unmittelbar aufeinander folgende Abschnittsmarken. Sie zeigen das logische Bandende an. Zwei aufeinander folgende Abschnittsmarken treten auch dann auf, wenn ein leerer Dateiabschnitt, oder eine leere Datei auf dem Band steht.
In diesem Fall werden sie nicht als Doppel-Abschnittsmarke interpretiert, sondern als zwei einfache Abschnittsmarken.

„Leer“ bedeutet: es gibt keine Datenblöcke zwischen den Abschnittsmarken, vor der Datei-Anfangskennsatzgruppe und nach der Datei-/Bandende-kennsatzgruppe.

FCB (File Control Block)

Dateisteuerblock, der die für die Dateiverarbeitung benötigten Informationen enthält.

First in – first out (FIFO)

Warteschlangen-Struktur, die Informationen in der Reihenfolge der Eingabe abarbeitet (im Gegensatz dazu: Last in – first out, LIFO).

FOREIGN-Datei

Eine FOREIGN-Datei ist eine Datei auf einem privaten Datenträger oder auf einem Net-Storage-Volume, die aber nicht auf einem Pubset katalogisiert ist.

geblockter Satz (blocked record)

Satz in einer Datei, in der jeder Datenblock mehrere Sätze oder Satzsegmente enthalten kann.

Kennsatz (Label)

Ein Satz am Beginn oder Ende eines Bandes oder einer Datei, der dazu dient, Band oder Datei zu identifizieren, zu beschreiben und/oder zu begrenzen. Ein Kennsatz wird nicht als Bestandteil der Datei betrachtet. Jeder Kennsatz wird für sich in einem eigenen Block aufgezeichnet (Kennsatzblock).

Kennsatzfamilie (Label Set)

Eine ununterbrochene Folge von Kennsätzen mit demselben Kennsatznamen.

Kennsatzgruppe (Label Group)

Eine ununterbrochene Folge von Kennsatzfamilien, die ein Band, einen Dateiabschnitt oder eine Datei begrenzt.

Kennsatzname (Label Identifier)
Ein Wort aus drei Zeichen, das als Teil des Kennsatzes aufgezeichnet ist und ihn kennzeichnet.

Kennsatzroutine (Label Handling Routine)
Folge von Anweisungen zur Verarbeitung von Kennsätzen.

Klasse-1-Speicher
Der Teil des virtuellen Speichers, der von den hauptspeicherresidenten Moduln des Ablaufteils belegt ist. Alle Seiten der Klasse 1 sind als privilegiert und nichtseitenwechselbar gekennzeichnet. Von diesen Seiten ist auf dem Seitenwechsel-Speicher kein Abbild enthalten. Die Seiten sind während des gesamten Systemlaufs im Hauptspeicher.

Klasse-5-Speicher
Der Teil des virtuellen Benutzerspeichers, der die für einen Benutzerauftrag erforderlichen seitenwechselbaren Bereiche enthält, die vom Ablaufteil dynamisch zugewiesen werden.

Klasse-6-Speicher
Der Teil des virtuellen Benutzerspeichers, der die Benutzerprogramme enthält, die vom Ablaufteil dynamisch zugewiesen werden.

Last Byte Pointer (LBP)
Zeiger auf das letzte gültige Byte des letzten logischen Blocks einer PAM-Datei.

Last Page Pointer (LPP)
Zeiger auf die letzte von einer Datei belegte PAM-Seite. Entspricht im Katalogeintragder Highest-Used-Page.

LBN (= Logical Block Number)
(laufende) Nummerierung der PAM-Seiten einer Datei

Locate-Mode
Im Locate-Mode fordert der Benutzer die Adresse des aktuellen Satzes an, der sich in einem Pufferbereich befindet. Für die Datenübertragung zum und vom Puffer ist der Benutzer zuständig.

mehrbenutzbare Datei

Eine Datei, die mit USER-ACCESS=ALL-USERS katalogisiert ist. Auf eine mehrbenutzbare Datei kann man von allen Benutzerkennungen aus zugreifen, sofern die übrigen Schutzattribute (z.B. Dateikennwörter) der Datei dies zulassen.

Move-Mode

Im Move-Mode gibt der Anwender die Lage des Satzes in seinem Programm an. Für die Datenübertragung zum und vom Puffer ist das System zuständig.

Net-Client

Realisiert den Zugriff auf Net-Storage für das nutzende Betriebssystem.
In BS2000 transformiert der Net-Client zusammen mit dem BS2000-Subsystem ONETSTOR die BS2000-Dateizugriffe in entsprechende UNIX-Dateizugriffe und führt sie über NFS auf dem Net-Server aus.
Net-Client bei SU /390 und S-Server ist der HNC, bei SU x86 und SQ-Server das Trägersystem X2000.

Net-Server

File-Server im weltweiten Rechnernetz, der Speicherplatz (Network Attached Storage, NAS) für die Nutzung durch andere Server bereitstellt und entsprechende File-Server-Dienste anbietet.

Net-Storage

Der von einem Net-Server im Rechnernetz bereitgestellte und zur Nutzung durch fremde Server freigegebene Speicherplatz. Net-Storage kann ein Dateisystem oder auch nur ein Knoten im Dateisystem des Net-Servers sein.

Net-Storage-Datei

Bezeichnet eine Datei, die auf einem Net-Storage-Volume angelegt ist. Auf Net-Storage wird zwischen den zwei Dateitypen BS2000-Datei und Node-File unterschieden.

Net-Storage-Volume

Net-Storage-Volumes repräsentieren Net-Storage im BS2000, die die Systembetreuung als Erweiterung von Daten-Pubsets bereitstellt.
Net-Storage-Volumes werden durch ihre Volume Serial Number (VSN) und den Volumetyp NETSTOR angesprochen. Die VSN des Net-Storage-Volumes entspricht im freigegebenen Dateisystem des Net-Servers dem Verzeichnis, das die Benutzerdateien und Metadaten enthält.

NFS (Network File System)

BS2000-Softwareprodukt, mit dem verteilte Datenhaltung in einem heterogenen Rechnernetz möglich ist. Der Benutzer kann auf ferne Dateien so zugreifen, als ob sie an seinem lokalen Rechner vorhanden wären.
NFS dient somit der Konnektivität zwischen Systemen. Außerdem können Dateien mit NFS automatisch und zuverlässig durch das BS2000 gesichert werden.

Node-File

Bezeichnet eine Net-Storage-Datei (FILE-TYPE=NODE-FILE), die sowohl von BS2000 als auch von offenen Systemen angelegt und bearbeitet werden kann. Node-Files werden ab BS2000 OSD/BC V10.0 unterstützt. Sie liegen auf einem Net-Storage-Volume in einem benutzerspezifischen Verzeichnis (Name der Benutzerkennung) und die Dateinamen entsprechen den BS2000-Namenskonventionen.

Nulldatei

Eine Datei, die logisch leer ist. Es handelt sich hier um eine Datei, die katalogisiert und der vom System Speicherplatz zugewiesen wurde, die jedoch keine Daten enthält.

Ortungsbetrieb

Im Ortungsbetrieb fordert der Benutzer die Adresse des aktuellen Satzes an, der sich in einem Pufferbereich befindet. Für die Datenübertragung zum und vom Puffer ist der Anwender zuständig.

PAM-Seite

Zusammenhängender Speicherplatz von 2048 Byte, beginnend an einer durch 2048 dividierbaren Adresse.

Privilegierter Modus/Programm

alle Teile des Betriebssystems, die nicht im unprivilegierten Verarbeitungszustand ablaufen.

Prozedur/Prozedurdatei

Datei, die eine festgelegte Kommando- oder Anweisungsfolge enthält, die der Programmeingabe dienen.
Prozeduren werden mit CALL-PROCEDURE bzw. ENTER-PROCEDURE gestartet. Nur wenn die Datei einen ENTER-Job enthält, wird sie mit ENTER-JOB gestartet. Näheres zu Prozeduren siehe Handbuch „Kommandos“ [3]).

Pubset

Satz gemeinschaftlich gekennzeichneter Platten. MPVS-Systeme arbeiten mit mehreren voneinander unabhängigen Pubsets.

Pubset-Id

Kennzeichen eines Pubsets. Beginnt die Archivnummer eines gemeinschaftlichen Datenträgers mit den drei Zeichen „PUB“, ist die Pubset-Kennung das 4. Zeichen, enthält sie einen Punkt, bilden die Zeichen vor dem Punkt die Pubset-Kennung. Im Pfadnamen wird die Pubset-Kennung in der Form :catid: (siehe Catid (= Katalogkennung)) angegeben.

Puffer

Zusammenhängender Speicherbereich: ein Teil des Hauptspeichers, aus dem Daten gelesen werden oder in den Daten geschrieben werden;

Public Volume Set (PVS)

veraltet für: Pubset; siehe Pubset

PVS-id

Siehe Pubset-Id

Satz

Eine Zusammenfassung von Daten, die als eine logische Einheit behandelt werden.

Satz fester Länge

Ein Satz in einer Datei, in der alle Sätze nach Vereinbarung dieselbe Länge haben;
innerhalb der Datei ist keine Anzeige der Länge erforderlich.

Satz variabler Länge

Satz in einer Datei, in der die Sätze unterschiedlich lang sein können. Die Satzlänge muss im ersten Wort innerhalb des Satzes angegeben werden. Dieses Satzlängenfeld wird bei der Ermittlung der Satzlänge mit einbezogen. Das Satzlängenfeld ist 4 Byte lang und enthält die Satzlänge linksbündig als Sedezimal- oder Dezimalzahl.

Satzformat

Festlegung einer Datei hinsichtlich der Länge und Segmentierung ihrer Sätze.

Satzformat V:	Wird ein Band des Satzformats V im Nicht-EBCDI-Code beschrieben, wird die Länge sedezimal angegeben.
Satzformat D:	Wird ein Band bei Satzformat D im ISO-7-Bit-Code beschrieben, wird die Länge dezimal 4-stellig angegeben.

SPOOLOUT

automatischer SPOOLOUT: automatische Ausgabe des Inhalts der Systemdatei SYSLST auf einen Drucker oder Versenden per E-Mail bei Auftragsende (EXIT-JOB bzw. LOGOFF).

Stapelbetrieb

Siehe Batchbetrieb

SYSFILE-Umgebung

Siehe Systemdateien;
Als SYSFILE-Umgebung kann die Gesamtheit der einem Auftrag zugewiesenen Systemdateien bezeichnet werden.

Systemdateien

einem Auftrag zugewiesene System-Ein-/Ausgabedateien.
Die (Standard-) Dateinamen SYSDTA, SYSSTMT, SYSCMD, SYSIPT, SYSLST, SYSLST01, SYSLST02, ..., SYSLST99, SYSOPT und SYSOUT bezeichnen vom Betriebssystem benutzte (System-) Dateien zur Daten- bzw. Kommandoeingabe an das Betriebssystem oder zur Datenausgabe durch das Betriebssystem. Diese Dateien werden jeweils durch die Task eingerichtet und bezeichnen vorgegebene Ein- bzw. Ausgabebereiche.
Der Anwender kann die primäre Zuordnung aufheben und den (Standard-) Dateinamen eigene katalogisierte Dateien bzw. zusammengesetzte S-Variablen (bei Einsatz des Software-Produkts SDF-P) zuweisen.
Ausführliche Informationen zu Systemdateien siehe Handbuch „Kommandos“ [3].

TFT (= Task File Table)

Tabelle, die mit dem Kommando ADD-FILE-LINK, Operand LINK-NAME, erstellt wird und aus der bei Dateieröffnung Datei- und Verarbeitungseigenschaften in den Dateisteuerblock übernommen werden.

TSN (= Task Sequence Number)
Siehe Auftragsnummer (= TSN, Task Sequence Number)

TST (Tape Set Table)
Tabelle, die die für einen Auftrag angeforderten Bänder anzeigt (in Zusammenhang mit TFT).

Übertragungsbetrieb
Im Übertragungsbetrieb gibt der Anwender die Lage des Satzes in seinem Programm an. Für die Datenübertragung zum und vom Puffer ist das System zuständig.

ungeblockter Satz (unblocked record)
Ein Satz in einer Datei, in der jeder Datenblock nur einen Satz oder ein Satzsegment enthalten darf.

VSEQ (= Volume Sequence Number)
Bezeichnet einen Dateiabschnitt bei Mehrbanddateien.

VTOC (= Volume Table of Contents)
Dateiverzeichnis im F1-Etikett einer privaten Platte oder auf einem Net-Storage-Volume.

Zugriffsmethode
Eine festgelegte Technik der Datenverwaltung, die dem Benutzer die Datenorganisa-tion sowie die Methode der Datenübertragung zwischen den Ein-/Ausgabegeräten und dem Arbeitsspeicher vorschreibt.

Zugriffsmethoden des DVS sind:

- EAM (Evanescent Access Method)
- SAM (Sequential Access Method)
- ISAM (Indexed Sequential Access Method)
- UPAM (User Primary Access Method)
- BTAM (Basic Tape Access Method)

Literatur

Die Handbücher finden Sie im Internet unter *http://manuals.ts.fujitsu.com*. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie auch in gedruckter Form bestellen.

[1] BS2000 OSD/BC
Einführung in das DVS
Benutzerhandbuch

[2] BS2000 OSD/BC
Makroaufrufe an den Ablaufteil
Benutzerhandbuch

[3] BS2000 OSD/BC
Kommandos
Benutzerhandbuch

[4] **SPOOL** (BS2000)
Benutzerhandbuch

[5] **DAB** (BS2000)
Disk Access Buffer
Benutzerhandbuch

[6] **RFA** (BS2000)
Remote File Access
Benutzerhandbuch

[7] BS2000 OSD/BC
Einführung in die Systembetreuung
Benutzerhandbuch

[8] **SECOS** (BS2000)
Security Control System - Zugangs- und Zugriffskontrolle
Benutzerhandbuch

[9] **ARCHIVE** (BS2000)
Benutzerhandbuch

[10] **HSMS** (BS2000)
Hierarchisches Speicher Management System
Benutzerhandbuch

[11] **HIPLEX MSCF** (BS2000)
BS2000-Rechner im Verbund
Benutzerhandbuch

[12] **Einführung in die XS-Programmierung (für ASSEMBLER-Programmierer) (BS2000)**
Benutzerhandbuch

[13] **PERCON** (BS2000)
Benutzerhandbuch

[14] BS2000 OSD/BC
Dienstprogramme
Benutzerhandbuch

[15] **DRV** (BS2000)
Dual Recording by Volume
Benutzerhandbuch

[16] BS2000 OSD/BC
Systeminstallation
Benutzerhandbuch

[17] **SDF-A** (BS2000)
Benutzerhandbuch

[18] **SDF-P** (BS2000)
Programmieren in der Kommandosprache
Benutzerhandbuch

[19] BS2000 OSD/BC
Dateien und Volumes größer 32 GB
Benutzerhandbuch

[20] **RSO** (BS2000)
Remote SPOOL Output
Benutzerhandbuch

[21] **JV** (BS2000)
Jobvariablen
Benutzerhandbuch

[22] **XHCS** (BS2000)
8-bit-Code-Verarbeitung im BS2000
Benutzerhandbuch

[23] BS2000 OSD/BC
System Managed Storage
Benutzerhandbuch

[24] **FUJITSU Server BS2000 SE Serie**
Bedienen und Verwalten
Benutzerhandbuch

[25] **openCRYPT** (BS2000)
Sicherheit mit Kryptographie
Benutzerhandbuch

[26] **VM2000**
Virtuelles Maschinensystem
Benutzerhandbuch

[27] **MAREN**
Band 1: Grundlagen der MBK-Verwaltung
Benutzerhandbuch

[28] **MAREN**
Band 2: Benutzerschnittstellen
Benutzerhandbuch

Stichwörter