limes datentechnik

FLAM-sub (MVS)

FRANKENSTEIN-LIMES-ACCESS-METHOD

BENUTZERHANDBUCH

— Ausgabe Mai 2014 —

Telefon (06172) 5919-0 ■ Telefax (06172) 5919-39
http://www.flam.de ■ http://www.limesdatentechnik.de

Benutzerhandbuch FLAM-sub V4.5

Vorwort

Das vorliegende Handbuch beschreibt die Handhabung des Subsystems FLAM unter dem Betriebssystem MVS, OS/390 und z/OS von IBM.

FLAM-sub stellt eine Schnittstelle zum Utility FLAM (MVS) dar und setzt eine Lizenz f r FLAM V4.x voraus.

Weiterf hrende Informationen zu FLAM finden Sie in dem Benutzerhandbuch FLAM (MVS), zur Verschl sselung in dem Handbuch FLAM & AES.

Inhaltsverzeichnis

_

~ nderungsprotokoll 7 FLAM-sub V4.5

Die vorliegende Version FLAM-sub V4.5 ist eine Anpassung an FLAM (MVS) V4.5. Zus tzlich ist folgende Neuerung enthalten:

- **KME=FKMEFILE**

  Diese Key Management Extension Routine liest einen Schl ssel aus einer sequentiellen Datei variabler oder fixer Satzl nge. Der Schl ssel kann in Syntax des CRYPTOKEY Parameters enthalten sein.

  Die Datei kann selbst wieder mittels RACF gesch tzt werden, so dass hier auf einfache Weise eine sichere Ver- und Entschl sselung durch das Subsystem erm glicht wird.

- **Lizenzpr fung**

  Das Subsystem wird jetzt generell mit FLAM (MVS) ausgeliefert, enth lt jetzt aber eine eigene Lizenzpr fung.

~ nderungsprotokoll 6 FLAM-sub V4.4

Die vorliegende Version FLAM-sub V4.4 ist eine Anpassung an FLAM (MVS) V4.4. Zus tzlich ist folgende Neuerung implementiert:

- **Unterst tzung der KME-Schnittstelle zum Anschluss an ein Key Management System**

  Hiermit ist es wie im FLAM-Utility m glich, ber ein Interface zu einem Schl sselverwaltungssystem Passworte/Schl ssel zu erhalten.

  Damit ist eine automatische Ver-/Entschl sselung m glich, ohne dass Passworte/Schl ssel im Protokoll zu sehen sind.

~ nderungsprotokoll 5 FLAM-sub V4.3

Die vorliegende Version FLAM-sub V4.3 ist eine Anpassung an FLAM (MVS) V4.3.

- **Unterst tzung der AES-Verschl sselung mittels Hardware**

  Mit Einf hrung von CPACF ist seit den Hardwaregenerationen z9 und z10 eine Verschl sselung mit Hilfe der Hardwareroutinen m glich. FLAM-sub erkennt selbstt tig diese M glichkeit und benutzt die neuen Routinen.

  Performance Steigerungen bis zu 30% der gesamten CPU-Zeit wurden im MODE=NDC (d.h. keine Komprimierung, nur Verpackung) realisert.

  Allgemein gilt: je mehr Daten zu ver-/entschl sseln sind, desto gr er ist die Einsparung von CPU-Zeit.

~ nderungsprotokoll 4 FLAM-sub V4.1

Die vorliegende Version FLAM-sub V4.1 ist eine Anpassung an FLAM (MVS) V4.1. Neue Funktionen wurden nicht implementiert. Das Handbuch wurde berarbeitet.

- **bessere Unterst tzung der AES-Verschl sselung**

  Durch eine verbesserte Implementierung des AES Algorithmus werden Performance Steigerungen bis zu 50 % gegen ber der alten Version erzielt.

~ nderungsprotokoll 3 FLAM-sub V4.0

Mit der vorliegenden Version FLAM-sub V4.0 ergeben sich einige Neuerungen gegen ber der Version 3:

- **Unterst tzung der AES-Verschl sselung (Advanced Encryption Standard)**

  Mit CRYPTOMODE=AES als Parametereingabe werden die Daten nach dem AES-Verfahren verschl sselt. Diese Methode wurde in FLAM (MVS) V4 eingef hrt. Zur Absicherung der Daten werden zus tzliche Kontrollfelder eingef gt, die ebenfalls mit AES gebildet werden (Hash-MACs).

  Die verwendete Verschl sselungsmethode ist im Komprimat gespeichert und muss bei der Entschl sselung nicht angegeben werden.

_

- **Splitten der FLAMFILE**

Es kann beim Erstellen einer Datei die zu erzeugende FLAMFILE seriell oder parallel geteilt (gesplittet) werden. Die Steuerung erfolgt ber Parameter (SPLITMODE, SPLITNUMBER, SPLITSIZE). Eine durch das FLAM-Utility gesplittete FLAMFILE wird auch durch das Subsystem korrekt bearbeitet (und umgekehrt).

Bei seriellem Splitt (SPLITMODE=SERIAL) wird nach Erreichen einer vorgegebenen Dateigr e (SPLITSIZE= *zahl in Megabyte*) die aktuelle Datei geschlossen und eine neue Datei erzeugt. Die Anzahl der so erzeugten Fragmente einer gesplitteten FLAMFILE ist nicht beschr nkt und h ngt nur von der erzeugten Datenmenge ab.

Bei parallelem Splitt (SPLITMODE=PARALLEL) wird in eine vorgegebene Zahl von Dateien (SPLITNUMBER= *zahl*) komprimiert. In der vorliegenden Version k nnen bis zu 4 Dateien erzeugt werden. Die Dateigr e ist von der Menge der erzeugten Komprimatsdaten abh ngig.

~ nderungsprotokoll 2 FLAM-sub V3.0

Mit der vorliegenden Version FLAM-sub V3.0 ergeben sich einige Neuerungen gegen ber der Version 2:

- **Unterst tzung der Kompressionsmethode ADC (Advanced Data Compression)**

Mit MODE=ADC als Parametereingabe werden die Daten nach dem ADC-Verfahren komprimiert. Wurden bisher mit MODE=CX8/VR8 beste Ergebnisse bei strukturierten Daten erzielt, erlangt man mit dieser Methode auch bei unstrukturierten Datens tzen sehr gute Komprimierungsergebnisse bei moderatem Ressourcenverbrauch.

- **Laden von VSAM-KSDS Dateien ber Utilities wie z.B. IEBGENER, SORT**

Das Subsystem stellt sich selbstt tig auf die Art der Ansteuerung ein, d.h. bei Vorlage eines Dateisteuerblocks (DCB) f r sequentielle Dateien werden die Daten auch nur als sequentiell im Komprimat beschrieben und eingestellt. Utilities erkennen eine Subsystemdatei nur als sequentielle Datei und sprechen sie damit auch nur als sequentielle Datei an, obgleich die Anwendung das Laden einer VSAM-KSDS Datei vorsieht.

Durch Parameterangabe k nnen f r FLAM-sub jetzt die Datens tze als indexsequentiell beschrieben werden, so dass das Komprimat entsprechend f r einen Direktzugriff als VSAM-KSDS aufgebaut werden kann, obgleich ein DCB beim OPEN auf die Datei vorgegeben wurde.

- **Starten des Subsystems als Started Task**

Zu Testzwecken kann jetzt FLAM-sub auch als Started Task aktiviert werden (S FLAM). Ein IPL ist dazu nicht mehr notwendig. Entsprechend kann es auch deaktiviert werden (P FLAM).

~ nderungsprotokoll 1 FLAM-sub V2.0

Mit der vorliegenden Version FLAM-sub V2.0 ergeben sich einige Neuerungen gegen ber der Version 1:

- **Unterst tzung von VSAM-Zugriffen**

Die aufrufenden Programme k nnen sowohl in Assembler als auch in COBOL geschrieben sein.

Es werden alle logischen VSAM-Zugriffe durch FLAM-sub V2.0 unterst tzt, sowohl im MOVE- oder LOCATE-Mode, als auch f r synchrone oder asynchrone Aufrufe.

Insbesondere werden VSAM-KSDS-Befehle auch f r den UPDATE-Fall folgerichtig umgesetzt.

Die Kompression und Minimierung auch der Index-Eintr ge bewirkt neben der Speicherplatz-Ersparnis eine Erh hung der Performance im Gesamtdurchsatz beim Zugriff auf VSAM-Dateien.

Physische Zugriffe ber Speicheradressen (RBA) werden nicht unterst tzt.

- **Unterst tzung von unkomprimierten Dateien zum Lesen**

Liegt als Eingabe eine unkomprimierte Datei vor, kann sie wahlweise ebenfalls ber das Subsystem gelesen werden.

Dazu ist der Schl sselwortparameter IG10 f r das Subsystem einzugeben:

//... DD ...,SUBSYS=(FLAM,IG10,'flam-parameter')

Bei komprimierten Dateien hat dieser Parameter keine Wirkung.

- TRACE-Funktion

Zu Testzwecken kann eine TRACE-Funktion eingeschaltet werden.

Dazu ist der Schl sselwortparameter TRACE f r das Subsystem einzugeben:

//... DD ...,SUBSYS=(FLAM,TRACE,'flam-parameter')

Es werden dann alle Funktionsaufrufe an FLAM sowie ACB und RPL des aufrufenden Programms in eine Datei protokolliert.

Die Protokolldatei muss in der JCL angegeben werden, der FLAM-Parameter 'MSGDDN=ddname' gibt den DD-Namen an. Ohne Angabe wird der Parameter der DEFAULT-Angabe (vgl. Job INST02 vom FLAM-Utility) entnommen (i.d.R. FLPRINT).

Die Protokolldatei kann wiederum als Subsystemdatei angegeben werden:

//FLPRINT DD DSN=tracedatei,SUBSYS=FLAM

- Neue Schnittstelle zum FLAM-Utility

Ab der vorliegenden Version werden alle ben tigten Module des FLAM-Utilities nachgeladen, es entf llt das Verkn pfen der Subsystemmodule mit dem Utility. Damit ist auch ein "Neulinken" bei Wechsel der Lizenznummer f r das Utility nicht mehr n tig.

Allerdings ben tigt die Bibliothek der Lademodule von FLAM-util jetzt ebenfalls die APF-Autorisierung.

Es wird empfohlen, die Lademodule des Subsystems und des Utilities in einer Bibliothek zu f hren.

1. FLAM als Subsystem

FLAM (MVS) als Utility komprimiert und verschl sselt Daten in eine Datei, der FLAMFILE. Diese Datei ist Betriebssystem bergreifend austauschbar und wird vom dortigen FLAM wieder dekomprimiert und entschl sselt.

FLAM (MVS) enth lt Interfaces, die von Anwendungsprogrammen genutzt werden k nnen. Damit ist eine flexible Anpassung an bestehende Aufgaben m glich, die vom Utility allein nicht bew ltigt werden k nnen.

W hrend bei einer Neuprogrammierung eine Benutzung des Interfaces sicher kein Problem darstellt, ist eine Einbettung in bestehende Programme immer mit Aufwand verbunden.

Hier setzt FLAM-sub ein.

FLAM-sub ist ein Subsystem des Betriebssystems und wird aktiviert durch die Subsystemschnittstelle der Kommandosprache (JCL) des Betriebssystems (MVS, OS/390, z/OS).

Alle Aufrufe eines Anwendungsprogramms zu Dateioperationen (wie z.B. ffnen, Schlie en einer Datei, Lesen und Schreiben von S tzen) werden durch FLAM-sub entgegen genommen und verarbeitet.

˜ nderungen sind in den aufrufenden Programmen in der Regel nicht notwendig, das aufrufende Programm bemerkt keinen Unterschied zur herk mmlichen Dateiverarbeitung.

ffnet das Anwendungsprogramm seine Datei, wird durch FLAM-sub eine FLAMFILE ge ffnet. Wird ein Datensatz geschrieben, komprimiert und verschl sselt FLAM-sub ihn in die FLAMFILE. Wird ein Datensatz gelesen, entnimmt ihn FLAM-sub der FLAMFILE und bergibt ihn dekomprimiert und entschl sselt dem Anwendungsprogramm. Schlie en der Originaldatei bewirkt ein Schlie en der FLAMFILE.

Es wird somit niemals die FLAMFILE komplett dekomprimert/entschl sselt sondern nur der gerade ben tigte Teil!

Mit Hilfe von FLAM-sub ist es sogar m glich, Programme mit Dateiformaten zu unterst tzen, f r die sie nicht geschrieben worden sind, d.h. Programm und Datenformate sind voneinander entkoppelbar.

Die vom Subsystem erzeugten Komprimatsdateien (FLAMFILEs) lassen sich vom FLAM-Utility jederzeit dekomprimieren, und alle vom FLAM-Utility erzeugten FLAMFILEs werden vom Subsystem FLAM akzeptiert und folgerichtig verarbeitet. Die gleiche Aussage trifft f r FLAMFILEs zu, die ber die Satzschnittstelle von FLAM erzeugt/gelesen werden (vgl. Handbuch zu FLAM V4.x).

2. Aufruf des Subsystems

FLAM-sub ist ein Subsystem des Betriebssystems und wird aktiviert durch die Subsystemschnittstelle der Kommandosprache (JCL) des Betriebssystems (MVS, OS/390, z/OS). Der Name des Subsystems ist FLAM.

```
//ddname     DD   DSN=dateiname,
//                SUBSYS=(FLAM,'flam-parameter')
```

Damit wird die katalogisierte Datei *dateiname* zur Bearbeitung mit dem Subsystem FLAM angewiesen.

Zus tzlich k nnen FLAM Parameter zum Ablauf mitgegeben werden, wie man es beim Utility FLAM gewohnt ist.

Fehlerhafte Parameter werden von FLAM als JCL-Fehler abgewiesen, der Job startet somit erst gar nicht. Die berpr fung des DD-Kommandos durch JES bleibt weiterhin erhalten.

Der Katalogeintrag der Datei wird durch FLAM nicht ver ndert. Alle Eintr ge wie Dateiname, Satz- und Blockl nge, Volume, usw. bleiben voll erhalten.

Damit unterliegen die mit dem Subsystem FLAM zu bearbeitenden Dateien weiterhin den Kontrollmechanismen wie SMS oder RACF.

Die Zahl der vom Subsystem "gleichzeitig" zu verarbeitenden Dateien ist seitens FLAM nicht beschr nkt. Eher wirken Systemgrenzen, wie z.B. der maximal zur Verf gung stehende Speicher oder die Zahl der m glichen DD-Statements.

Sollen Dateien bearbeitet werden, bei denen die SUBSYS-Angabe im DD-Statement nicht unterst tzt wird (wie z.B. JES-Dateien), kann durch Angabe eines FLAM-Parameters und eines zweiten DD-Statements das Problem umgangen werden:

```
//ddname     DD   SUBSYS=(FLAM,'FLAMDDN=ddname1'),
//                DCB=(LRECL=....,BLKSIZE=....)
//ddname1    DD   SYSOUT=G,DEST=(.....),
//                DCB=(LRECL=80,.....)
```

Durch Angabe eines DD-Namens *ddname1* als FLAM Parameter wird das Subsystem veranla t, die durch *ddname1* zugewiesene Datei zu bearbeiten (hier: zu beschreiben). Diese muss in der JCL angegeben sein. Analog k nnte *ddname1* als Eingabe zur Dekomprimierung dienen. Das aufrufende Programm dagegen benutzt die mit ddname zugewiesene Datei!

Mit dieser Technik kann jede Datei (JES, RJE, weiteres Subsystem, Magnetb nder, tempor re Dateien, ...) schreibend oder lesend bearbeitet werden, die im ersten Fall sonst nicht behandelt werden k nnte (vgl. die Beispielsammlung).

Werden f r ddname DCB-Attribute mitgegeben, so interpretiert FLAM sie als Werte f r die originale (unkomprimierte) Datei. DCB-Angaben f r *ddname1* gelten f r das Komprimat, der FLAMFILE.

Damit lassen sich v llig unterschiedliche Satz- und Blockformate f r die Original- und Komprimatsdatei einstellen, wie es das Problem erfordert. D.h. eine Applikation kann mit dem Subsystem jetzt Dateien benutzen, die mit den Eintr gen im Programm nichts mehr gemein haben (siehe Beispielsammlung) und ggf. ohne Subsystem konvertiert werden m ten.

Wird die FLAMFILE ohne SUBSYS-Angabe angesprochen, verh lt sie sich wie eine "normale" Datei. D.h. sie kann durch Utilities oder File-Transfer-Programme gelesen (kopiert, transferiert) werden, ohne dass es einer Dekomprimierung bedarf!

Analog dazu kann auch eine z.B. vom File Transfer bertragene FLAMFILE ber das Subsystem gelesen und verarbeitet werden.

3. Arbeitsweise

Bevor ein Batch-Job von JES gestartet wird, werden alle Angaben der JCL berpr ft. Dabei werden die Parameter f r das Subsystem bergeben und von FLAM-sub gepr ft. Im Fehlerfall wird das DD-Statement als JCL-Fehler abgewiesen. Zus tzlich wird eine Meldung in der JCL-Liste ausgegeben (vgl. Beispielsammlung; Meldungen). Der Job wird somit nur bei Fehlerfreiheit gestartet und wird andernfalls wie gewohnt abgebrochen.

Durch Aufruf des OPEN-Befehls im Programm wird die Verbindung zum Subsystem FLAM hergestellt. Ob komprimiert oder dekomprimiert werden soll, entscheidet FLAM-sub zu diesem Zeitpunkt. F r Open Input wird nur gelesen und dabei dekomprimiert und ggf. entschl sselt, f r Open Output wird nur geschrieben, komprimiert und ggf. verschl sselt, Open I/O erlaubt vollen I/O auf die FLAMFILE ber den Schl ssel der Originals tze. Beim OPEN werden Dateityp, Dateiorganisation, Satz- und Blockl nge dem Programm des Aufrufers (oder den DCB-Angaben in der JCL) entnommen und bei Open Output zus tzlich in den FLAM Fileheader der FLAMFILE bertragen.

Zum Open-Zeitpunkt werden auch die in der JCL angegebenen FLAM-Parameter aktiv und die Komprimierungs-/Dekomprimierungsroutinen entsprechend eingestellt.

Es sind alle Zugriffe (GET, WRITE, PUT, POINT, ERASE, CHECK) mit entsprechenden RPL-Modifikationen bei VSAM ber das Subsystem m glich.

Alle Lese- und Schreibzugriffe werden gem den Angaben im Programm befolgt.

Asynchrone VSAM-Aufrufe werden vom Subsystem synchron durchgef hrt, Returncodes aber erst beim CHECK-Aufruf zur ckgeliefert (bzw. die Fehlerroutinen angesprungen).

Es ist nicht von Bedeutung, ob die Datei als BSAM, QSAM oder VSAM vom Programm angesprochen wird. FLAM-sub bildet alle Aufrufe auf die vorliegende FLAMFILE ab.

Das Satzformat kann in allen F llen variabel oder fix sein.

Im Fehlerfall bleibt die Fehlerroutine im aufrufenden Programm aktiv und wird nicht durch das Subsystem berlagert. Die vom Subsystem zur ckgegebenen Fehler-Returncodes entsprechen denen des DMS (siehe auch 'MVS/DFP Macro Instructions for VSAM Data Sets' f r VSAM Fehlercodes), d.h. die Fehlerroutinen m ssen nicht ge ndert werden; die IEC... Meldungen des DMS (MVS Message Library: System Messages) werden ggf. weiterhin protokolliert. Bei FLAM-Fehlern und Fehlern des Subsystems werden zus tzlich Meldungen mittels WTO, Routcde=11 durch FLAM-sub ausgegeben (vgl. Meldungen; Beispielsammlung). Sie werden sowohl im Systemlog als auch in der JCL-Liste dokumentiert.

Ansonsten werden keine Protokolle erstellt (der FLAM-Parameter des Utilities, SHOW=ALL, bleibt wirkungslos).

Beim CLOSE auf die Datei schlie t FLAM die FLAMFILE und gibt alle beim OPEN angeforderten Arbeitsbereiche wieder frei. Ein erneuter OPEN-Aufruf ist erlaubt.

FLAM-sub arbeitet im Adressraum des Anwenders. Ein eventueller Fehler oder Abbruch eines Programms, Dateizugriffs o. . hat keinen Einflu auf Subsystem-Zugriffe anderer Prozesse und Anwender.

Der Dateiname *name* des DD-Statements (DD DSN=*name*,SUBSYS=FLAM) wird beim Komprimieren (OPEN OUTPUT) mit in das Komprimat bernommen. Das kann durch den FILEINFO-Parameter ggf. ausgeschaltet werden.

4. Voraussetzungen zum Einsatz

FLAM muss sowohl als Utility (ab V4.0) als auch als Subsystem auf dem aktuellen Rechner installiert sein (vgl. Installationsanweisung).

Es k nnen je nach verwendetem Dateityp Datens tze geschrieben und gelesen, je nach verwendetem Komprimierungsmodus auch ge ndert, eingef gt oder gel scht werden. Die Zugriffe m ssen logisch, d.h. ber die sequentielle Reihenfolge oder ber einen Schl ssel erfolgen. Zugriffe ber die RBA (relative byte address) eines Datensatzes oder ber Alternate Index sind nicht m glich !

Sind im Anwendungsprogramm keine Dateiattribute definiert (wie z.B. Satz- oder Blockgr e), m ssen diese Angaben im DD-Statement definiert werden (siehe Beispielsammlung).

Im aufrufenden Programm kann ein DCB f r PS-Dateien oder ein ACB f r VSAM-Dateien codiert worden sein.

Runtime Systeme (von z.B. COBOL, PL/I, LE) berpr fen oftmals zugeordnete Dateien, die bearbeitet werden sollen. Dann muss die FLAMFILE dem Typ der Originaldatei entsprechen, d.h. bei Zugriff auf eine VSAM-ESDS Datei muss auch die FLAMFILE eine VSAM-ESDS Datei sein. Ist die originale Datei vom Typ VSAM-KSDS, so ist auch die FLAMFILE als KSDS anzulegen.

Eine FLAMFILE im KSDS-Format muss wie im FLAM-Utility aufgebaut sein (siehe Handbuch zu FLAM (MVS)), d.h.

Relative Key Position	0
Keyl nge	1 Byte l nger als das Original
Satzl nge	zwischen 80 und 32760 Bytes
Control Interval Size	beliebig je nach Satzl nge

Die Satzl nge der VSAM-KSDS-Datei kann zwar beliebig eingestellt werden und FLAM kann sich entsprechend darauf einstellen, es sind aber ein paar Gesichtspunkte zu beachten, die die Performance unmittelbar betrifft:

FLAM komprimiert je nach MAXR- und MAXB-Angabe eine Menge von Datens tzen ‘in einem Rutsch’. D.h. dass z.B. bei MAXB=64 eine Datenmenge von 64 KB komprimiert wird. Bei einer Komprimierungsrate von 90% erg be es eine Komprimatsl nge von ca. 6,4 KB ! Hatte ihre Originaldatei z.B. eine Satzl nge von 400 Byte und Sie w rden auch die FLAMFILE so anlegen, m te FLAM 17 VSAM-S tze schreiben, um dieses Komprimat wegschreiben zu k nnen. Damit m ten auch 17 S tze gelesen werden, um einen Originalsatz dekomprimieren zu k nnen. Der Durchsatz w re wesentlich h her, wenn die FLAMFILE mit einer mittleren Satzl nge von ca. 7KB angelegt worden w re. Dann m te pro ‘Matrix’ auch nur ein VSAM-Satz gelesen oder geschrieben werden.

Hinweis:
Programm FLAMCKV der FLAM (MVS) Auslieferung unterst tzt die Analyse einer VSAM-KSDS FLAMFILE (siehe Handbuch zu FLAM (MVS) V4.3 und neuer).

5. Einschr nkungen

Allgemein:

- AES-Verschl sselung einer VSAM-KSDS Datei ist nicht zul ssig.
- MODE=ADC ist im Updatemode nicht zul ssig.
- Alle direkt ber FLAM-sub zugewiesenen Dateien (// DD DSN=*name*,... ,SUBSYS=FLAM) m ssen katalogisiert sein.
- Dateigenerationen k nnen nicht mit relativen Namen angesprochen worden (wie z.B. DATEI(+3)) sondern es muss der absolute Dateiname angegeben werden (z.B. DATEI.G0008V00). Ggf. kann ein zweites DD-Statement angegeben werden:

```
//ddname   DD  SUBSYS=(FLAM,'FLAMDDN=KOMPRI')
//KOMPRI   DD  DSN=DATEI(+3),
               DISP=(NEW,CATLG),
               UNIT=...,SPACE=...
```

- Der Parameter DISP des DD-Statements (wie z.B. DISP=NEW oder DISP=-(OLD, DELETE) kann nicht ausgewertet werden und hat somit keine Wirkung.

  Sollen Dateien neu angelegt werden, so kann das ber ein zweites DD-Statement erfolgen:

```
//ddname   DD  SUBSYS=(FLAM,'FLAMDDN=NEUDAT')
//NEUDAT   DD  DSN=...,
               DISP=(NEW,CATLG),
               UNIT=...,SPACE=...
```

  oder die Datei muss in einem fr heren STEP angelegt worden sein.

- Der Parameter AMP darf nicht angegeben sein.

  Falls weiterhin notwendig ist er beim zweiten DD-Statement anzugeben.

Es sind alle Zugriffe (GET, WRITE, PUT, POINT, ERASE, CHECK) mit entsprechenden RPL-Modifikationen bei VSAM ber das Subsystem m glich.

Folgende Einschr nkungen gelten (siehe auch Handbuch DFSMS Macro Instructions for Data Sets):

DCB-Aufrufe (PS-Dateien):

- Werden in Programmen Angaben wie Satz- und Blockl nge weggelassen (Utilities wie IEBGENER oder SORT arbeiten so), m ssen DCB-Angaben im DD-Statement gemacht werden!

  Die bliche Technik, dass in diesem Fall beim OPEN die Werte aus einem Katalogeintrag automatisch bernommen werden, funktioniert bei Einsatz eines Subsystems nicht. Eine OPEN-Routine erkennt nur, dass eine Subsystemdatei vorliegt und greift dann nicht mehr auf einen (sogar vorhandenen) Katalogeintrag zu.

- Schreiben im LOCATE Mode wird abgewiesen.

- Ein tempor rer Close-Aufruf bei BSAM-Zugriffen (CLOSE ...,TYPE=T) wird nicht zum Subsystem weitergereicht und hat keine Wirkung.

- POINT bei BSAM-Zugriffen wird mit der Fehlermeldung IEC141I 013-BC .. abgewiesen (wird erst gar nicht zum Subsystem durchgereicht).

- Eine Datei mit RECFM=VS oder VBS (spanned records) l t sich zwar beschreiben, aber ber die Subsystemschnittstelle nicht mehr lesen (Fehlermeldung IEC141I 013-A8 ...).

ACB/RPL-Aufrufe (VSAM-Dateien):

- Zugriffe mit mehreren RPL's (RPL-Liste) werden nicht unterst tzt (es wird nur die erste RPL ausgewertet).

- Asynchrone Aufrufe werden synchron von FLAM-sub ausgef hrt (aber die Reaktion auf Fehler erfolgt erst bei CHECK wie gewohnt).

- CNVTAD gibt ung ltige Angaben zur ck, bzw. Null.

- ENDREQ (Terminate a Request) bleibt ohne Wirkung (FLAM kennt keine Sperren und h lt nur den letzten Key).

- VERIFY (Synchronize End of Data) bleibt ohne Wirkung (VSAM-Steuerbl cke werden nicht aktualisiert).

_

- SHOWCB ACB=...
  Die Felder CINV, KEYLEN, LRECL, RKP geben die Werte der originalen Datei wieder, alle anderen Angaben (wie z.B. ENDRBA, HALCRBA, ...) beziehen sich auf die FLAMFILE.

Soll eine VSAM-KSDS-Datei ber FLAM-sub geladen werden (OPEN OUTPUT im Programm auf eine VSAM-KSDS-Datei), m ssen dem Subsystem Angaben ber Schl sselposition und -l nge mitgegeben werden (Achtung: die Schl sselposition ist bei FLAM gegen ber der RKP bei IDCAMS um 1 Byte erh ht !), beim Laden ber Utilities auch der Dateityp KSDS:

```
// DD  ..,SUBSYS=(FLAM,'OKEYP=wert1,OKEYL=wert2,ODSORG=KSDS')
```

Das hat folgenden Grund:

blicherweise werden diese Angaben bei der Katalogisierung mittels IDCAMS mitgegeben. Bei VSAM-Zugriffen enth lt nicht der ACB diese Angaben sondern der Katalogeintrag. Hier ist aber die FLAMFILE mit v llig anderen Werten katalogisiert worden, somit stehen die Parameter der Originaldatei nicht mehr zur Verf gung. Bei sp teren Zugriffen (INPUT oder I/O) stehen diese Angaben im Komprimat und m ssen nicht mehr angegeben werden.

6. Parameter f r FLAM

Parameter f r FLAM werden im DD-Statement bei der SUBSYS-Angabe bergeben:

SUBSYS=(FLAM,**'parameter1=wert1,parameter2=wert2,...'**)

oder auch

SUBSYS=(FLAM,**'parameter1=wert1','parameter2=wert2','...'**)

Die Parameter entsprechen denen des FLAM Utilities (vgl. Handbuch FLAM V4.x).

Statt des Gleichheitszeichens '=' k nnen auch Klammern () verwendet werden, wie z.B. MO(VR8).

Enthalten Parameterwerte ein Hochkomma, so sind zwei Hochkommata anzugeben (wie z.B. SUBSYS=(FLAM,'CRYPTOK=C''PASS WORD''').

CRYPTOKEY
CRYPTOK

Schl ssel zur Ver- bzw. Entschl sselung des Komprimats
Mit der Angabe des Schl ssels wird das eingestellte Verschl sselungsverfahren aktiviert (siehe Parameter CRYPTOMODE).

M gliche Werte

1 - 64 Zeichen in der Form A'...', C'...', X'...' oder als String

Bei A'...' werden die Zeichen gem der internen FLAM Tabelle E/A in ASCII umkodiert.

Standard: kein Schl ssel

G ltig f r: Komprimierung, Dekomprimierung

Hinweis: Bitte beachten Sie, dass abdruckbare Zeichen nationaler Zeichens tze auch bei der Dekomprimierung identisch (d.h. bin r-kompatibel) eingegeben werden m ssen. F r heterogenen Austausch empfehlen wir die hexadezimale Eingabe X'...'.

CRYPTOMODE
CRYPTOM

Art des Verschl sselungsverfahrens
M gliche Werte:

AES	Advanced Encryption Standard
FLAM	das interne FLAM Verfahren

Standard: FLAM

G ltig f r: Komprimierung.

Hinweis: AES wurde mit FLAM V4.0 eingef hrt und ist in lteren Versionen nicht entschl sselbar.

Die Verschl sselung wird erst durch Angabe eines Schl ssels (Parameter CRYPTOKEY) aktiviert. Das Verschl sselungsverfahren ist im Komprimat gespeichert und muss zur Dekomprimierung nicht angegeben werden.

Verschl sselung setzt MODE=ADC oder NDC voraus. Ohne Angabe des Kompressionsmodus wird ADC eingestellt.

FILEINFO
FI

Dateinamen des Originals in Fileheader bernehmen.

M gliche Werte:

YES	Dateinamen bernehmen
NO	Dateinamen nicht bernehmen

Standard: YES

G ltig f r: Komprimierung

FLAMDDN
FLAMD

DD-Name einer durch JCL zugewiesenen FLAMFILE.

Damit wird in diese Datei komprimiert bzw. aus dieser dekomprimiert.

M gliche Werte:

DD-Name bis max. 8 Zeichen

Standard: kein Name

G ltig f r: Komprimierung, Dekomprimierung

Hinweis: Der Dateiname (DSN=*name*) des DD-Statements des Subsystems wird bei der Komprimierung in die FLAMFILE bernommen, nicht der, auf den FLAMDD zeigt.

HEADER
HE

Einen Fileheader erzeugen

M gliche Werte:

YES	Fileheader erzeugen
NO	keinen Fileheader erzeugen

Standard: YES

G ltig f r: Komprimierung

Hinweis: Der Header sollte stets erzeugt werden, damit das Subsystem bei sp terer Verarbeitung darauf zugreifen kann. HEADER=YES wird bei AES-Verschl sselung automatisch gesetzt.

KMEXIT

KME

Anwendungsprogramm zur Schl sselverwaltung bei Ver-/Entschl sselung aktivieren.

M gliche Werte:

name	Name des Moduls (max. 8 Zeichen)

Standard: kein Exit

G ltig f r: Verschl sselung, Entschl sselung

Der Modul wird dynamisch geladen.

Hinweis: Hiermit kann ein Schl ssel zur Ver-/Entschl sselung bereitgestellt werden (siehe Kap. 3.5.5 im FLAM Handbuch).

Dieser Schl ssel berschreibt eine evtl. CRYPTOKEY-Angabe.

Achtung: Diese Anwendungsprogramme m ssen in Assembler geschrieben sein! Programme in Hochsprachen mit einem Runtime-System (wie Cobol, C, ...) f hren zu einem Systemfehler.

KMPARM

Parameter f r den KMEXIT

KMP

Diese Parameter werden an das Anwendungsprogramm zur Schl sselverwaltung bergeben (siehe Kapitel 3.5.5 KMEXIT im FLAM Handbuch).

M gliche Werte:

1 – 256 Zeichen in der Form A'...', C'...', X'...' oder als String

Bei A'...' werden die Zeichen gem der internen FLAM-Tabelle E/A (siehe Anhang im FLAM Handbuch) in ASCII umkodiert.
Standard: keine Parameter

G ltig f r: Verschl sselung, Entschl sselung

.

MAXBUFFER
MAXB

Maximale Gr e der Komprimatsmatrix.

M gliche Werte:

0 - 7 (Kompatibilit t zu FLAM V2.0)

8 - 2047 Gr e in Kilobyte

Standard: 64 KByte

G ltig f r: Komprimierung

Hinweis: Nur einstellbar f r die Komprimierungsmodi CX7/CX8/VR8, ADC und NDC verwenden stets 64 KB.

MAXRECORDS
MAXR

Maximale Anzahl S tze in einer Komprimatsmatrix.

M gliche Werte:

1 - 255 f r MODE=CX7/CX8/VR8
1-4095 f r MODE=ADC

Standard: 255, bzw. 4095 je nach MODE-Angabe

G ltig f r: Komprimierung

Hinweis: f r FLAMFILEs im KSDS-Format:
Wird berwiegend direkt zugegriffen, sollte MAXR kleiner gew hlt werden (<= 64).

MAXSIZE
MAXS

Maximale Satzl nge der Komprimatsdatei.

M gliche Werte:

80 - 32760

Standard: 512 Byte

G ltig f r: Komprimierung

Hinweis: Bei fixen Dateien (RECFM=F, FB, FBS) hat LRECL des Katalogs Vorrang, bei variablen Dateien und bei VSAM wird MAXS als max. L nge benutzt, auch wenn LRECL oder RECSIZE im Katalog gr er definiert sind.

MODE
MO

Komprimierungsvariante

M gliche Werte:

ADC	8-Bit Komprimat h chster Effizienz
CX7	transformierbares 7-Bit Komprimat
CX8	8-Bit Komprimat (laufzeitoptimiert)
VR8	8-Bit Komprimat (Speicherplatz optimiert)
NDC	keine Komprimierung

Standard: VR8

G ltig f r: Komprimierung

Hinweis: Bei Verschl sselung wird automatisch MODE=ADC gesetzt.

MSGDDN
MSGD

DD-Name einer durch JCL zugewiesenen Protokolldatei.
In diese Datei wird ein TRACE-Protokoll (siehe Parameter TRACE) geschrieben.

M gliche Werte:

DD-Name bis max. 8 Zeichen

Standard: FLPRINT

G ltig f r: Komprimierung, Dekomprimierung

ODSORG
ODSO

Dateiorganisation einer Ausgabedatei

M gliche Werte:

PS, KSDS

Standard: PS, d.h. Erstellen einer sequentiellen Datei

G ltig f r: Komprimierung

Dieser Parameter ist nur n tig, wenn eine VSAM-KSDS-Datei geladen werden soll, obgleich ein OPEN auf eine sequentielle Datei erfolgt (z.B. bei IEBGENER, SORT).

OKEYLEN
OKEYL

Schl ssell nge der Originaldatei bei Ausgabe

M gliche Werte:

0, 1 - 255

Standard: 8

G ltig f r: Komprimierung

OKEYPOS
OKEYP

Schl sselposition der Originaldatei bei Ausgabe

M gliche Werte:

0, 1 bis Satzl nge minus Schl ssell nge

Standard: 1

G ltig f r: Komprimierung

SECUREINFO
SEC

Zusatzinformationen in der FLAMFILE, die die Sicherheit erh hen (Verriegeln der FLAMFILE, Manipulationsschutz). Jede Ver nderung an dieser FLAMFILE f hrt zum Abbruch der Dekomprimierung.

M gliche Werte:

YES	Erzeugen dieser Informationen (Standard bei Verschl sselung)
NO	Keine zus tzlichen Informationen speichern
IGNORE	Beim Dekomprimieren Fehler durch Verletzung dieser Sicherheitsinformationen ignorieren

Standard: NO (ohne Verschl sselung)
YES (mit AES-Verschl sselung)

G ltig f r: Komprimierung mit MO=ADC/NDC, Dekomprimierung

Hinweis: Bei AES-Verschl sselung wird automatisch SECURE=YES gesetzt.
Verletzungen k nnen z.B. entstehen durch Konkatinieren mehrerer so gesicherter FLAMFILES, durch unbemerkte Abbr che eines Filetransfers (z.B. bei FTP), durch Manipulation, durch Updatefunktionen.

SPLITMODE
SPLITM

Art des Splittens einer Komprimatsdatei
M gliche Werte:

NONE	kein Splitt
SERIAL	serieller Splitt
PARALLEL	paralleler Splitt

Standard: NONE

G ltig f r: Komprimierung

Hinweis: Splitting von FLAMFILEs wurde in FLAM (MVS) V4.0 eingef hrt und ist mit lteren Versionen nicht zu bearbeiten.

Die Information ist im Komprimat gespeichert und muss zur Dekomprimierung nicht angegeben werden.

Datei- oder DD-Namen m ssen Ziffernfolgen im Namen haben (siehe Handbuch FLAM (MVS) V4.x Kapitel 3.1.5, Beispiel in Kapitel 5.1.3 oder hier Beispiel 10.8).

Bei parallelem Splitt m ssen ALLE Dateien katalogisiert sein und in der JCL angegeben werden, die Zuordnung muss ber den DD-Namen erfolgen (z.B. FLAMDD=CMP01).

SPLITNUMBER
SPLITN

Anzahl paralleler Splitts
M gliche Werte:

2 - 4	Anzahl ‚gleichzeitig' zu schreibender Dateien

Standard: 4

G ltig f r: Komprimierung

Hinweis: Die Information ist im Komprimat gespeichert und muss zur Dekomprimierung nicht angegeben werden.

Bei Dekomprimierung m ssen alle Dateien gleichzeitig im Zugriff sein. Einzelne Dateien k nnen nicht dekomprimiert werden.

Dieser Parameter setzt SPLITMODE=PARALLEL voraus.

SPLITSIZE
SPLITS

Splittgrenze in MB bei seriellem Splitt
M gliche Werte:

1 - 4095

Standard: 100

G ltig f r: Komprimierung

Hinweis: Die Zahl der insgesamt erzeugten Dateien ist von der Datenmenge abh ngig. Sie ist im Komprimat gespeichert und muss zur Dekomprimierung nicht angegeben werden.

Dieser Parameter setzt SPLITMODE=SERIAL voraus.

TRANSLATE
TRA

Daten-Konvertierung

M gliche Werte:

E/A	Konvertiert von EBCDIC nach ASCII
A/E	Konvertiert von ASCII nach EBCDIC
name	Name (max. 8 Zeichen) eines Daten-Moduls, der eine 256 Byte lange bersetzungstabelle enth lt.

Standard: keine Konvertierung

G ltig f r: Komprimierung oder Dekomprimierung (aber nicht gleichzeitig, d.h. I/O-Modus)

7. Parameter zur Steuerung des Subsystems

Parameter zur Steuerung des FLAM Subsystems sind Schl sselwortparameter, die separat von FLAM-Parametern angegeben werden m ssen. Sie dienen nur der Steuerung des Subsystems und haben mit FLAM im eigentlichen Sinne nichts zu tun.

...SUBSYS=(FLAM,**parm**,'flam-parameter')

IG10 Ignorieren des Returncodes 10 (keine FLAMFILE) beim ffnen zum Lesen. Es werden die S tze ohne Dekomprimierung an den Aufrufer bergeben.

Dieser Parameter ist in Systemen sinnvoll, in denen sowohl komprimierte als auch unkomprimierte Dateien vorliegen k nnen (z.B. bei Tests).

TRACE Einschalten der TRACE-Funktion.

Es werden alle Funktionsaufrufe an FLAM sowie die Kontrollbl cke (ACB, RPL) des Aufrufenden Programms protokolliert.

Die Protokolldatei muss in der JCL angegeben werden, der FLAM-Parameter 'MSGDDN=ddname' gibt den DD-Namen an (ohne Angabe wird in die Datei FLPRINT geschrieben). Ist die Datei nicht in der JCL zugeordnet, wird keine Fehlermeldung ausgegeben sondern ohne TRACE gearbeitet.

8. Meldungen des Subsystems

FLAM-sub gibt nur im Fehlerfall an der Konsole eine Meldung aus (mittels WTO ,ROUTCDE=11). Sie wird ebenfalls im JCL-Log protokolliert.

Ein Buchstabe nach der Meldungsnummer gibt den Zeitpunkt der Fehlererkennung an:

I	bei der Initialisierung
C	bei der Analyse der JCL
A	bei der Datei-Allokation
O	beim Er ffnen einer Datei
D	w hrend des I/O-Vorgangs
E	beim Schlie en einer Datei

Au er bei der Initialisierung des Subsystems betreffen Fehler nur die angegebene Datei des jeweiligen Jobs, d.h. das Subsystem bleibt aktiv f r die anderen Dateien.

FLM0500I INITIALIZATION OF SUBSYSTEM FLAM COMPLETED

Das Subsystem FLAM wurde korrekt initialisiert und steht zur Verf gung.

Reaktion: keine

FLM0501I SUBSYSTEM FLAM NOT INSTALLED ON THIS SYSTEM

Das Subsystem konnte nicht initialisiert werden. Es wurde auf dem Rechner nicht installiert.

Reaktion: in SYS1.PARMLIB(IEFSSN..) den Eintrag FLAM,FLSSIPL erg nzen oder mit S FLAM die Started Task aktivieren.

FLM0502I SUBSYSTEM FLAM ALREADY INITIALIZED

Das Programm zur Initialisierung des Subsystems wurde nach dem IPL erneut gestartet und wurde abgewiesen. Das bereits aktive Subsystem bleibt t tig.

Reaktion: ein erneuter Start von FLAM-sub ist nicht notwendig

FLM0503I NO MEMORY RECEIVED FOR INITIALIZATION

Das Betriebssystem hat keinen Arbeitsbereich zur Initialisierung bereitstellen k nnen.
Das Subsystem ist nicht aktiv.

Reaktion: eine Analyse ist erforderlich (Systemdump)

_

FLM0504I SUBSYSTEM COULD NOT FREE SYSTEM MEMORY

Der vom Betriebssystem erhaltene Arbeitsbereich konnte nicht freigegeben werden.
Das Subsystem wurde aber initialisiert und steht bereit.

Reaktion: eine Analyse ist erforderlich (Systemdump)

FLM0505I SUBSYSTEM FLAM FUNCTION MODULE NOT FOUND

Zur Initialisierung des Subsystems fehlt ein Modul aus FLAM-sub. Eine vorige Betriebssystemmeldung benennt den Modul. Das Subsystem ist nicht initialisiert.

Reaktion: die Subsystemmodule in einer Bibliothek der Masterkennung (SYS1.) einstellen und mit der LINKLIB verketten (vgl. Installationsanweisung).

FLM0506I MODULE IEFJSVEC NOT FOUND

Es fehlt der Betriebssystemmodul IEFJSVEC zur Initialisierung. Das Subsystem ist nicht initialisiert.

Reaktion: fehlenden Modul in eine mit der LINKLIB verketteten Bibliothek einstellen.

FLM0507I IEFJSVEC: NO MEMORY FOR SSVT

Der Betriebssystemmodul IEFJSVEC hat die Initialisierung des Subsystems wegen Speichermangels abgebrochen.

Reaktion: Analyse erforderlich (Systemdump)

FLM0508I LOGIC ERROR IN IEFJSVEC

Der Betriebssystemmodul IEFJSVEC hat die Initialisierung des Subsystems wegen eines internen Fehlers abgebrochen.

Reaktion: Analyse erforderlich (Systemdump)

FLM0510I INTERNAL ERROR. RC = nr

Es wurde ein interner Fehler erkannt. Das Subsystem wurde nicht initialisiert.

Reaktion: bitte verst ndigen Sie Ihren Vertriebspartner

_

FLM0515I INITIALIZATION OF SUBSYSTEM FLAM FAILED

Das Subsystem konnte wegen eines Fehlers nicht initialisiert werden. Der Fehler wurde in einer vorherigen Meldung protokolliert.

FLM0519I FLAM SUBSYSTEM IS NOT LICENSED

Die Lizenz umfasst nicht die Nutzung des Subsystems. Bitte informieren Sie Ihren Vetriebspartner.

FLM0520C PARAMETER ERROR: ...
FLM0502C SYNTAX ERROR: ...

Es wurde im DD-Statement der JCL ein fehlerhafter Parameter f r FLAM-sub eingegeben.

Reaktion: Parameter korrigieren (siehe: Parameter) und den Job neu starten.

FLM0523C NO MEMORY RECEIVED FOR JCL-CONVERTION

Ein Subsystemmodul hat keinen Arbeitsspeicher vom Betriebssystem erhalten.

Reaktion: Analyse erforderlich (Systemdump)

FLM0530A PARAMETER ERROR: ...
FLM0530A SYNTAX ERROR: ...

Es wurde im DD-Statement der JCL ein fehlerhafter Parameter f r FLAM-sub eingegeben.
Die fehlerhaften Parameter werden angezeigt.

Reaktion: Parameter korrigieren (siehe: Parameter) und den Job neu starten.

FLM0531A NO MEMORY RECEIVED FOR ALLOCATION

Ein Subsystemmodul hat keinen Arbeitsspeicher vom Betriebssystem erhalten.

Reaktion: Analyse erforderlich (Systemdump)

_

FLM0540O OPEN ERROR. DDNAME=*ddname*. RC= nr (FLAM)/(VSAM)

F r die mit *ddname* in der JCL zugewiesenen Datei wurde zum Open-Zeitpunkt ein Fehler erkannt. FLAM-Returncodes werden dezimal angegeben, VSAM-Fehler sedezimal (vgl. Manual FLAM-utility; oder z.B. DFSMS/MVS Macro Instructions for Data Sets, SC26-4913)

Dieser Meldung folgt die IEC141I 013-C0 Message des Data Management Systems, wobei der vom Programm verwendete DD-Name ausgegeben wird. Bei VSAM-Zugriffen wird ggf. die Message IEC161 ausgegeben.

Die FLAM-Fehlercodes entsprechen dem Utility, siehe Kapitel 8 im Handbuch zu FLAM (MVS).

Hier die wichtigsten:

nr:

-1 Speichermangel (evtl. Lizenzschutzverletzung)
10 Datei ist keine FLAMFILE (bei Input oder I/O)
11 FLAMFILE Formatfehler
12 Satzl ngenfehler
13 Dateil ngenfehler
14 Checksummenfehler
21 Unzul ssiger Matrixpuffer
22 Unzul ssiges Kompressionsverfahren
23 Unzul ssiger CODE in FLAMFILE
24 Unzul ssiger BLOCKMODE
25 Unzul ssige Satzl nge
29 Fehlender oder falscher Crypto-Schl ssel
31 Datei nicht zugewiesen (DD-Statement fehlt)
33 Ung ltiger Dateityp
34 Ung ltiges Satzformat
35 Ung ltige Satzl nge
36 Ung ltige Blockl nge
40 Modul oder Tabelle (TRANSLATE-Parameter!) nicht ladbar
60-78 FLAM-Syntaxfehler
120 Beim Teilen oder Zusammenf gen einer FLAMFILE kann kein weiterer Dateiname (oder DD-Name) gebildet werden. Z.B. m te nach Datei Nummer 9 die Nummer 10 generiert werden, der Name enth lt aber nur eine einstellige Ziffernfolge.
121 Beim Zusammenf gen einer gesplitteten FLAMFILE fehlt ein Fragment (Datei).
122 Beim Zusammenf gen einer seriell gesplitteten FLAMFILE liegen die Fragmente (Dateien) in falscher Reihenfolge vor.
123 Fragmente (Dateien) einer gesplitteten FLAMFILE geh ren nicht zusammen.
124 Eine FLAMFILE wurde in mehr Dateien geteilt, als die aktuelle Version zusammen f gen kann.
125 Formatfehler im letzten Satz einer parallel gesplitteten FLAMFILE.
130 Eine mit SECURE=YES komprimierte FLAMFILE ist beim Lesen nicht mehr im originalen Zustand.
Kann z.B. durch ˜ nderung, Erg nzung (Konkatinierung) geschehen sein, dann ggf. mit SECURE=IGNORE dekomprimieren.

131 In einer mit SECURE=YES komprimierten FLAMFILE fehlen beim Lesen Datens tze (z.B. auch fehlendes oder unvollst ndiges Member in einer Sammel-FLAMFILE).
Falls erlaubt, mit SECURE=IGNORE dekomprimieren.

132 In eine mit SECURE=YES komprimierten Sammel- FLAMFILE wurde ein Member eingef gt. Wird beim Dekomprimieren erkannt.
Falls erlaubt, mit SECURE=IGNORE dekomprimieren.

133 Die Reihenfolge der S tze einer mit SECURE=YES komprimierten FLAMFILE wurde ver ndert.
Falls erlaubt (z.B. wegen Update), mit SECURE=IGNORE dekomprimieren.

134 Eine mit SECUREINFO=NO komprimierte FLAMFILE enth lt Security-Informationen.
Dieser Fehler l t sich nicht ignorieren. Im einfachsten Fall wurden FLAMFILEs ohne und mit Security-Informationen konkatiniert, dann muss die Konkatinierung aufgehoben werden.

531 bei parallelem Splitt sind nicht alle Fragmente (Dateien) in der JCL zugeordnet

Reaktion: bei RC = -1 und korrekter Lizenzierung von FLAM die REGION-Angabe im EXEC-Statement vergr ern, bei RC = 11 - 14 liegt die Datei nicht im urspr nglichen Zustand vor, der Dateninhalt wurde ver ndert (evtl. durch File-Transfer-Programme), ansonsten gem der Erkl rung Fehler beheben (siehe auch Manual zu FLAM-utility, Kapitel 8).

FLM0541O ALLOCATION FAILED FOR DSN *dateiname* SVC99 ERROR CODE = *nr1*. INFO CODE = *nr2*.

Bei der dynamischen Allokation der Datei *dateiname* wurde ein Fehler erkannt. Die Codes sind sedezimal angegeben.

Reaktion: Error- und Info Code des SVC99 (Dynamic Allocation) gem Handbuch (z.B. 'MVS Authorized Assembler Services Guide, GC28-1763') analysieren.

z.B. **Error Code = 1708**: die angegebene **Datei ist nicht katalogisiert** (FLAM-sub kann direkt nur auf katalogisierte Daten zugreifen)

FLM0543O TRANSLATE-PARAMETER INVALID WITH OPEN I/O - IGNORED

Der TRANSLATE-Parameter wird bei OPEN I/O ignoriert.
Nur bei OPEN INPUT oder OPEN OUTPUT k nnen Eingabe- oder Ausgabes tze gem der angegebenen Tabelle umgesetzt werden.

Die Verarbeitung wird ohne Datenumsetzung weitergef hrt.

_

FLM0544O FORCED BY PARAMETER TO ACCESS AS UNCOMPRESSED DATA

Beim ffnen einer Subsystemdatei wurde keine FLAMFILE erkannt (Meldung FLM0440O mit RC=10), der gesetzte Subsystemparameter IG10 erzwingt eine weitere Verarbeitung, bei der eine unkomprimierte (Original-) Datei angenommen wird.

Achtung: liegt keine Originaldatei vor, kann es zu nicht vorhersagbaren Fehlern f hren.

FLM0552D I/O ERROR. DDNAME = *ddname*. RC = *nr* (FLAM)/(VSAM) dateiname

Ein Fehler wurde vom Subsystem FLAM w hrend des Lesens oder Schreibens der angegebene Datei *ddname* erkannt. Der Returncode ist dezimal f r FLAM-Returncodes angegeben, Security-Verletzungen werden sedezimal ausgegeben, VSAM-Returncodes sind sedezimal dargestellt.

Alle FLAM-Returncodes sind im Handbuch zu FLAM (Kapitel 8) beschrieben.

VSAM-Returncodes finden Sie z.B. im Handbuch 'DFSMS/MVS Macro Instructions for Data Sets, SC26-4913'.

Der Meldung folgt ggf. die IEC020I 001-... Message des DMS.

nr:

11 FLAMFILE Formatfehler
12 Satzl ngenfehler
13 Dateil ngenfehler
14 Checksummenfehler

Bei o.a. Returncodes liegt eine Datenverf lschung der FLAMFILE vor (z.B. durch File Transfer).

7 Passwort ist nicht angegeben
15 Satzl nge gr er als 32 KB
16 Satzl nge gr er als MAXB - 4
25 Unzul ssige Satzl nge
29 Passwortfehler

120-134 siehe FLM0540O.

Werden bei berpr fung der Security Informationen Fehler festgestellt, so wird der Fehlercode sedezimal (nnmmmm) ausgegeben. Nn bezeichnet den Fehlerort, mit nn =

01 Header
02 Segment
03 Membertrailer
04 Filetrailer

Mit mmmm wird der Fehler selbst beschrieben:

0001 MAC1, Mac ber das Komprimat
0002 MAC2, Verkettungs MAC
0004 MAC3, Mac ber Macs
0010 Daten fehlen
0020 Daten eingef gt
0040 Daten aktualisiert (update)
0080 Satzz hler Komprimat
0100 Bytez hler Komprimat
0200 Satzz hler Originaldaten
0400 Bytez hler Originaldaten
0800 Verkettung bei FLAM-Verschl sselung

Es k nnen mehrere Fehler gleichzeitig gemeldet werden.
Z.B. besagt der Fehlercode 030180, dass sowohl die Anzahl Komprimatss tze (0080) als auch die Anzahl der Komprimatsbytes (0100) nicht mit den gespeicherten Werten bereinstimmen.

Um die FLAMFILE bei Verletzung der Security evtl. trotzdem dekomprimieren zu k nnen, kann SECUREINFO=IGNORE angegeben werden.

FLM0553D NO MEMORY RECEIVED FOR I/O OPERATIONS

Ein Subsystemmodul hat keinen Arbeitsspeicher vom Betriebsystem erhalten. Es k nnen dadurch auch keine weiteren Angaben gemacht werden.

Reaktion: vergr ern Sie die REGION-Angabe im EXEC-Statement.

FLM0570E CLOSE ERROR. DDNAME = ddname. RC = nr dateiname

Ein Fehler wurde beim Schlie en der angegebenen Datei erkannt.

Da beim Schlie en einer Datei noch evtl. Komprimatss tze geschrieben werden m ssen vgl. Meldung FLM0552D.

9. Installation von FLAM-sub

Das Subsystem ist in der Auslieferung von FLAM (MVS) enthalten. Es bedarf aber einer g ltigen Lizenz zur Nutzung, ohne die das Subsystem FLAM nicht startet.

9.1 LINKLST und Autorisierung

Das Subsystem l dt bei der Initialisierung seine Module und zum OPEN-Zeitpunkt Module des FLAM-Utilities nach, daher muss die LOAD-Bibliothek mit der Systembibliothek SYS1.LINKLST verkettet und APF-autorisiert sein.

Es gibt mehrere Wege, Verkettung und Autorisierung zu erreichen (vgl. z.B. z/OS MVS Initialization and Tuning Reference).

In beiden F llen sind Eintr ge in Membern der Bibliothek SYS1.PARMLIB notwendig, xx ist die laufende Nummerierung Ihrer aktiven Systemgenerierung.

9.1.1 Statische Methode

Dieser Weg ist gerade in lteren Systemversionen verbreitet. Zum Systemstart sind alle Bibliotheken definiert, der Zustand ist statisch und erwartet w hrend eines Systemlaufs keine Ver nderung.

Zur Verkettung mit der Systembibliothek ist

im Member LNKLSTxx die Zeile

SYS1.FLAMLIB,

einzuf gen.

In der Regel sind die so konkatinierten Bibliotheken standardm ig APF-autorisiert. Andernfalls ist zur Autorisierung

im Member IEAAPFxx einzuf gen:

SYS1.FLAMLIB *volume*,

mit *volume* als Name der Platte, auf der die Bibliothek SYS1.FLAMLIB gespeichert ist (ist dieser Eintrag der letzte im Member, entf llt das Komma).

9.1.2 Dynamische Methode

Diese wird in neueren Betriebssystemversionen verwendet und erlaubt eine flexiblere Handhabung und ˜ nderungen w hrend eines Systemlaufs.

Das Member PROGxx erh lt neue Eintr ge. Das Member l t eine gro e Anzahl von Varianten zu, deshalb hier nur als Beispiel:

Zur Verkettung

```
LINKLST     ADD
            NAME(LNKLSTxx)
            DSNAME(SYS1.FLAMLIB)
            VOLUME(volume)
```

Und ggf. zur Autorisierung

```
APF         ADD
            DSNAME(SYS1.FLAMLIB)
            VOLUME(volume)
```

9.2 Start des Subsystems

9.2.1 Statische Methode

Diese Methode l t eine ˜ nderung des Subsystems w hrend eines Systemlaufs nicht zu. Damit bleibt w hrend der gesamten Zeit FLAM-sub im beim IPL definierten Zustand, eine Deaktivierung oder Aktualisierung ist nicht m glich.

Das Subsystem FLAM wird w hrend eines System-IPL's initialisiert. Dazu ist im Member IEFSSNxx der Bibliothek SYS1.PARMLIB eine Zeile einzuf gen:

```
FLAM,FLSSIPL
```

Der Eintrag darf nicht vor den Zeilen mit JES, RACF oder SMS vorgenommen werden.

Die Reihenfolge der Eintr ge richtet sich danach, wie h ufig die Programme angesprochen werden. Je fter sie aufgerufen werden, desto eher sollten die Eintr ge im ersten Drittel des Members erfolgen.

Ggf. muss der Eintrag in der Form

```
SUBSYS SUBNAME(FLAM) INITRTN(FLSSIPL)
```

erfolgen, weil die anderen Subsysteme des IEFSSNxx Members so eingetragen sind (Mischeintr ge sind nicht erlaubt). Das Subsystem bleibt statisch.

Nach dem n chsten System-IPL steht das Subsystem FLAM zur Verf gung.

Der Start des Subsystems auch per Konsol-Kommando erfolgen:

```
SETSSI ADD,S=FLAM,I=FLSSIPL
```

9.2.2 Dynamische Methode

Im Rechenzentrum mit einem 24 X 7 Stunden-Betrieb ist die statische Methode sicherlich zu unflexibel.

FLAM-sub kann deshalb als Started Task von der Konsole aktiviert werden:

S FLAM

Die zugeh rige Prozedur liegt der Auslieferung bei.

Wird die Befehlszeile

COM='S FLAM'

in das Member COMMNDxx in SYS1.PARMLIB eingef gt, wird FLAM-sub beim IPL des Betriebssystems automatisch gestartet und muss nicht extra an der Konsole eingegeben werden.

Mit dem Konsolaufruf

P FLAM

wird FLAM-sub deaktiviert.

Mit

F FLAM,VER

k nnen die Programmversionen der Subsystemmodule an der Konsole ausgegeben und somit berpr ft werden.

Nach einer Deaktivierung lie en sich z.B. Module einer neueren FLAM-sub Version installieren, die dann (nach einem LLA REFRESH auf die Bibliothek) durch ein erneutes S FLAM aktiviert werden k nnten.

10. Beispielsammlung

10.1 Ein-/Ausgabe mit katalogisierten Dateien

Die Datei USER.BEISPIEL.EINGABE ist als Komprimatsdatei (FLAMFILE) erstellt worden (z.B. mit FLAM-Utility). Sie soll jetzt von einem anderen Programm gelesen werden. Die Datei USER.BEISPIEL.AUSGABE wurde bisher von diesem Programm erstellt und soll jetzt ebenfalls komprimiert werden. Beide Dateien sind katalogisiert.

Dazu sind zwei F lle zu unterscheiden.

a) Das Programm hat zum Open-Zeitpunkt der Datei die Satz- und Blockl nge im ACB oder DCB (oder in der FD Section bei COBOL) angegeben.

L sung: es gen gt die SUBSYS-Angabe im DD-Statement.

```
//stepname EXEC PGM=programm
//EINGABE  DD   DSN=USER.BEISPIEL.EINGABE,
//              DISP=SHR,
//              SUBSYS=FLAM
//AUSGABE  DD   DSN=USER.BEISPIEL.AUSGABE,
//              DISP=OLD,
//              SUBSYS=FLAM
```

Da im aufrufenden Programm die "richtigen" Werte angegeben sind, werden die Datens tze vom Subsystem gem dieser Angaben bergeben bzw. bernommen.

b) Das aufrufende Programm geht von einem "originalen" Katalogeintrag aus, d.h. im Programm sind weder Satz- noch Blockl nge angegeben (IEBGENER, SORT und viele Dienstprogramme arbeiten so).

L sung: zus tzliche Angabe von DCB-Attributen im DD-Statement.

```
//stepname EXEC PGM=programm
//EINGABE  DD   DSN=USER.BEISPIEL.EINGABE,
//              DISP=SHR,
//              DCB=(RECFM=FB,LRECL=121,BLKSIZE=3025),
//              SUBSYS=FLAM
//AUSGABE  DD   DSN=USER.BEISPIEL.AUSGABE,
//              DISP=OLD,
//              DCB=(RECFM=VB,LRECL=438,BLKSIZE=4096),
//              SUBSYS=FLAM
```

Dem Programm werden unabh ngig vom tats chlichen Katalogeintrag die angegebenen DCB-Attribute vorgegeben. Das Subsystem bergibt oder bernimmt S tze gem diesen Angaben. FLAM dagegen liest oder schreibt Komprimatss tze im katalogisierten Format auf die Platte.

Im JCL-Listing erkennt man die Subsystemaktivit t an den Systemmeldungen.

```
IEF237I FLAM ALLOCATED TO EINGABE
IEF237I FLAM ALLOCATED TO AUSGABE
   .
   .
   .
IEF285I   USER.BEISPIEL.EINGABE           SUBSYSTEM
IEF285I   USER.BEISPIEL.AUSGABE           SUBSYSTEM
```

10.2 FLAM-Parameter und Subsystem

Wie im Beispiel 1. Zus tzlich soll f r die Ausgabedatei die Kompressionsmethode ADC (Advanced Data Compression) eingestellt werden.

L sung:

Die dazu notwendigen Parameter werden in der SUBSYS-Angabe des DD-Statements angegeben.

```
//stepname EXEC PGM=programm
//EINGABE  DD   DSN=USER.BEISPIEL.EINGABE,
//              DISP=SHR,
//              SUBSYS=FLAM
//AUSGABE  DD   DSN=USER.BEISPIEL.AUSGABE,
//              DISP=OLD,
//              SUBSYS=(FLAM,'MO=ADC')
```

Mit FLAM-Utility und den Parametern 'D,SHOW=DIR' oder der OPTION I in den FLAM-Panels kann man sich vom Erfolg der Komprimierung berzeugen.

10.3 Neuerstellung einer Datei

Die Datei USER.BEISPIEL.NEUDAT soll erzeugt werden. Sie ist nicht katalogisiert:

```
//AUSGABE  DD  DSN=USER.BEISPIEL.NEUDAT,
//             DISP=(NEW,CATLG,),
//             UNIT=SYSDA,SPACE=(CYL,(24,24)),
//             DCB=(RECFM=FB,LRECL=121,BLKSIZE=3025)
```

L sung: Angabe eines zweiten DD-Statements f r FLAM-sub.

Die originale Dateibeschreibung erh lt einen neuen DD-Namen, auf den ein FLAM-Parameter bei der SUBSYS-Angabe hinweist.

```
//AUSGABE  DD  SUBSYS=(FLAM,'FLAMDD=NEUDAT'),
//             DCB=(RECFM=FB,LRECL=121,BLKSIZE=3025)
//NEUDAT   DD  DSN=USER.BEISPIEL.NEUDAT,
//             DISP=(NEW,CATLG,),
//             UNIT=SYSDA,SPACE=(CYL,(6,6)),
//             DCB=(RECFM=FB,LRECL=121,BLKSIZE=3025)
```

Damit wird das Komprimat USER.BEISPIEL.NEUDAT genau wie das Original angelegt. Der Vorteil besteht darin, dass evtl. folgende Kopier- und Transferprogramme, die ohne die SUBSYS-Angabe arbeiten, den gleichen Katalogeintrag vorfinden wie beim Original und nicht angepa t werden m ssen.

Das neue DD-Statement ben tigt die DCB-Angabe, da im Original die Angabe ebenfalls erforderlich war.

Andernfalls kann das Komprimat beliebig eingestellt werden, z.B.:

```
//AUSGABE DD  SUBSYS=(FLAM,'FLAMDD=NEUDAT'),
//            DCB=(RECFM=FB,LRECL=121,BLKSIZE=3025)
//*
//NEUDAT  DD  DSN=USER.BEISPIEL.NEUDAT,
//            DISP=(NEW,CATLG),
//            UNIT=SYSDA,SPACE=(CYL,(6,6)),
//            DCB=(RECFM=FB,LRECL=1024,BLKSIZE=0)
```

Dem aufrufenden Programm gegen ber (und auch dem Subsystem) sind die Originals tze 121 Byte lang.

Die FLAMFILE dagegen erh lt aus Performancegr nden eine gr ere Satz- und Blockl nge.

Die Dateibeschreibungen sind somit voneinander entkoppelbar.

Diese Vorgehensweise ist zu empfehlen und sollte die Regel sein !

10.4 Tempor re Dateien

FLAM-sub kann direkt keine tempor ren Dateien bearbeiten. Dazu ist ein weiteres DD-Statement erforderlich.

```
//DDNAME   DD   SUBSYS=(FLAM,'FLAMDD=tempdat')
//tempdat  DD   DSN=&&DATEI,DISP=...
```

Die Angabe eines Dateinamens ist wegen der FLAMDD-Angabe f r DDNAME nicht notwendig. Es wird direkt auf die tempor re Datei verzweigt.

10.5 Andere Subsysteme

Andere Subsysteme, wie z.B. JES f r SYSOUT/SYSIN, m ssen ber ein weiteres DD-Statement angesprochen werden.

```
//DDNAME    DD    SUBSYS=(FLAM,'FLAMDD=subdat')
//subdat    DD    SYSOUT=A,DEST=(....),...
```

Unter Umst nden m ssen je nach verwendetem Programm DCB-Parameter f r die Verarbeitung der Originaldatei (DDNAME-Statement) angegeben werden (vgl. Beispiel 1).

10.6 Laden einer VSAM-KSDS Datei

Laden einer KSDS-Datei hei t, die (VSAM-)Datei zum Schreiben (OUTPUT) zu er ffnen. Dabei m ssen die Schl ssel in aufsteigender Reihenfolge an VSAM bergeben werden (wird auch durch FLAM-sub gepr ft).

Laden im I/O Modus ist m glich, d.h. die Datei enth lt mindestens einen Satz, verschlechtert aber die Performance und erh ht die Anzahl der Zugriffe.

Die Datei selbst wird in der Regel mittels IDCAMS erstellt und ist beim Jobablauf bereits katalogisiert.

Am einfachsten wird eine komprimierte KSDS-Datei mit dem FLAM-Utility erzeugt.

10.6.1 Laden als VSAM-KSDS

Hier wird die Subsystemdatei als VSAM-KSDS Datei angesprochen, d.h. es wurde ein ACB f r diese Datei aufgebaut.

Beim Laden ber das Subsystem muss die Schl sselbeschreibung der Originaldaten als Parameter mitgegeben werden:

```
//AUSGABE  DD  DSN=USER.BEISPIEL.AUSGABE,
//             SUBSYS=(FLAM,'OKEYPOS=21,OKEYLEN=64,MAXS=7168'),
//             DCB=(LRECL=512,BLKSIZE=20480)
```

Die Originals tze sind hier max. 512 Byte lang, das Controlintervall belegt 20480 Byte, der Schl ssel beginnt auf Position 21 und ist 64 Byte lang.

Das Subsystem soll die max. Satzl nge der VSAM-Datei ausnutzen (7168 Byte).

Hinweis: FLAM z hlt die Schl sselposition von 1 an (entspricht RKP=0).

Die FLAMFILE-VSAM-Datei USER.BEISPIEL.AUSGABE selbst ben tigt die Eintr ge (vgl. Kapitel 'Voraussetzungen zum Einsatz') f r das Beispiel

KEYS(65 0)
RECSZ(7168 7168)

Achtung: auch bei gr erer RECSZ-Angabe wird durch den MAXS-Parameter die FLAMFILE-Satzl nge auf 7168 Byte begrenzt.

Anmerkung:

Zum Laden einer VSAM-Datei muss diese leer sein, d.h. sie darf keinen Satz enthalten, oder der Parameter REUSE zum berschreiben der Datei muss bei IDCAMS angegeben worden sein.

_

10.6.2 Laden ber Utilities

Die meisten Utilities (wie z.B. SORT, IEBGENER) stellen sich selbstt tig auf die Ausgabedatei ein, d.h. eine sequentielle (PS-)Datei wird mittels eines DCBs angesprochen, eine VSAM-KSDS-Datei ber einen ACB.

Eine Subsystemdatei wird in der Regel nur als sequentielle (PS-)Datei erkannt und angesprochen, d.h. es wird ein DCB f r diese Datei aufgebaut.

Normalerweise w rde FLAM-sub diese Daten sequentiell (d.h. ohne Schl ssel-Kenntnis) ablegen und mit einem PS-Eintrag versehen.

Parameter ‘ bersteuern’ diese Vorgehensweise, so dass die Daten als VSAM-KSDS abgelegt werden k nnen. Bei Vorlage einer VSAM-KSDS-FLAMFILE kann somit eine Komprimatsdatei f r Direktzugriffe erstellt werden.

Beim Laden ber das Subsystem muss die Schl sselbeschreibung der Originaldaten und die Dateiorganisation KSDS als Parameter mitgegeben werden:

```
//AUSGABE  DD  DSN=USER.BEISPIEL.AUSGABE,
// SUBSYS=(FLAM,'ODSORG=KSDS,OKEYPOS=21,OKEYLEN=64,MAXS=7168'),
// DCB=(LRECL=512,BLKSIZE=20480)
```

Die Originals tze sind hier max. 512 Byte lang, das Controlintervall belegt 20480 Byte, der Schl ssel beginnt auf Position 21 und ist 64 Byte lang.

Das Subsystem soll die max. Satzl nge der VSAM-Datei ausnutzen (7168 Byte).

Hinweis: FLAM z hlt die Schl sselposition von 1 an (entspricht RKP=0).

Die VSAM-KSDS-FLAMFILE selbst ben tigt die Eintr ge (vgl. Kapitel ’Voraussetzungen zum Einsatz’) f r das Beispiel

KEYS(65 0)
RECSZ(7168 7168)

Anmerkung:

FLAM-sub berpr ft beim Laden die aufsteigende Reihenfolge der Schl ssel und gibt im Fehlerfalll den entsprechenden VSAM-Returncode zur ck. Die Utilities geben diesen Fehlerfall in der Regel nur als allgemeinen „WRITE-ERROR“ wieder.

10.7 TRACE-Funktion

Zu Testzwecken wird sowohl f r die Eingabe- als auch die Ausgabedatei ein Trace eingeschaltet:

```
//EINGABE   DD  SUBSYS=(FLAM,TRACE)
//AUSGABE   DD  DSN=dateiname1,
//              SUBSYS=(FLAM,TRACE,'MSGDDN=FLTRACE')
//*
//FLPRINT   DD  SYSOUT=*
//FLTRACE   DD  DSN=dateiname3,DISP=(NEW,CATLG),
              SPACE=(TRK,(12,12),RLSE),UNIT=SYSDA
```

Es m ssen unterschiedliche Dateien f r den Trace angegeben werden. FLTRACE ist die Zuordnung f r den Trace der Datei AUSGABE, FLPRINT wird, da ohne Parametereingabe im DD-Statement, als Standardname f r die Datei EINGABE angenommen.

Das Protokoll f r die EINGABE sieht dann wie folgt aus (die S tze sind hier verk rzt):

```
FLAM - TRACE FUNCTION -   COPYRIGHT 1995-2012 BY LIMES DATENTECHNIK GMBH

 ACB DURING OPEN: 009E13A0
 (A000004C 00000000 00000000 54000000 00000000 00000000 48900008 0000000
  00000000 00000000 002C0041 009E12A8 03000000 0000521C 00000000 0C30005
  00000000 00000000 00000000)

 FLMOPN, ON ENTRY:
 1:03700C04(037008A0) 2:03700C08(8349A192) 3:0349AA90(00000001) 4:0349AA
 FLMOPN, ON RETURN:
 1:03700C04(0372CD88) 2:03700C08(00000000) 3:0349AA90(00000001) 4:0349AA
 FLMOPD, ON ENTRY:
 1:03700C04(0372CD88) 2:03700C08(00000000) 3:0349AA90(00000001) 4:03700C
 5:03700CC4()
 6:03700C20(00000000) 7:03700C24(00000009) 8:03700C28(00000200) 9:03700C
 10:03700C5C(00000000 00000000
 00000001 00000008 00000000 00000000 00000000 00000000 00000000 000000000
 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000000000
 11:03700C34(00001800) 12:03700C38(00000000) 13:03700C3C(00000000)
 FLMOPD, ON RETURN:
 1:03700C04(0372CD88) 2:03700C08(00000000) 3:0349AA90(00000001) 4:03700CC
 5:03700CC4()
 6:03700C20(00000000) 7:03700C24(00000008) 8:03700C28(00000100) 9:03700CC
 10:03700C5C(00000000 00000000
 00000000 00000000 00000001 00000000 00000000 00000000 00000000 000000000
 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000000000
 11:03700C34(00001800) 12:03700C38(00000000) 13:03700C3C(00000000)
 FLMOPF, ON ENTRY:
 1:03700C04(0372CD88) 2:03700C08(00000000) 3:03700C40(00000000) 4:03700C4
 7:03700C50(00000001) 8:03700C54(000000FF) 9:03700C5C(00000000 00000000
 00000001 00000008 00000001 00000000 00000000 00000000 00000000 000000000
 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000000000
 10:03700C58(00000001) 11:03700CC4(        ) 12:03700CC(        )
```

```
                                                                               _

FLMOPF, ON RETURN:
1:03700C04(0372CD88) 2:03700C08(00000000) 3:03700C40(000000C8) 4:03700C4
7:03700C50(00000001) 8:03700C54(000000FF) 9:03700C5C(00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
10:03700C58(00000001) 11:03700CC4(        ) 12:03700CC(        )
FLMGHD, ON ENTRY:
1:03700C04(0372CD88) 2:03700C08(00000000) 3:03700E54(00000036)
4:03700E58(FLAM.DAT.CMP                                              )
5:03700E4C(00000000) 6:03700E48(00000009) 7:03700E3C(00000050) 8:03700E0
9:03700C5C(00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
10:03700E40(00000C30) 11:03700E38(00000000) 12:03700E28(01010000)
FLMGHD, ON RETURN:
1:03700C04(0372CD88) 2:03700C08(00000000) 3:03700E54(0000001C)
4:03700E58(FLAM.FLAMV27C.CLIST(ORGFLAM))
5:03700E4C(00000000) 6:03700E48(00000009) 7:03700E3C(00000050) 8:03700E0
9:03700C5C(00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
10:03700E40(00000C30) 11:03700E38(00000000) 12:03700E28(01010000)
I/O-REQUEST: 00000000 GET
ACB: 000052F0
(A000004C 000147C0 8003E170 54000000 00000000 00000000 48900008 0000000
 00000000 00000000 002C0041 009E12A8 12000000 0000521C 00000000 0C30005
00000000 00000000 00000000)
RPL: 00005390
(0000004C 00000000 00000000 00000000 00000000 00000000 000052F0 0000000
 00045E5C 00000000 20000000 00000000 00000050 00000050 00000000 0000000
 00000000 00000000 000053EC)
FLMGET, ON ENTRY:
1:03700C04(0372CD88) 2:03700C08(00000000) 3:000053C0(00000050)
4:00045E5C(
5:000053C4(00000050)
FLMGET, ON RETURN:
1:03700C04(0372CD88) 2:03700C08(00000000) 3:000053C0(00000050)
4:00045E5C(PROC 0
5:000053C4(00000050)
I/O-REQUEST: 00000000 GET
ACB: 000052F0
(A000004C 000147C0 8003E170 54000000 00000000 00000000 48900008 000000
 00000000 00000000 002C0041 009E12A8 12000000 0000521C 00000000 0C3000
 00000000 00000000 00000000)
RPL: 00005390
(0000004C 00000000 00000000 00000000 00000000 00000000 000052F0 000000
 00045EAC 00000000 20000000 00000000 00000050 00000050 00000000 000000
 00000000 00000000 000053EC)
FLMGET, ON ENTRY:
1:03700C04(0372CD88) 2:03700C08(00000000) 3:000053C0(00000050)
4:00045EAC(
5:000053C4(00000050)
```

```
FLMGET, ON RETURN:
1:03700C04(0372CD88) 2:03700C08(00000000) 3:000053C0(00000050)
4:00045EAC(CONTROL NOLIST NOSYMLIST NOCONLIST NOFLUSH NOMSG NOCAPS
5:000053C4(00000050)
        .
        .
FLMCLS, ON ENTRY:
1:03700C04(0372CD88) 2:03700C08(00000002)
FLMCLS, ON RETURN:
1:03700C04(FFFFFFFF) 2:03700C08(00000000)
```

Die Namen der FLAM-Funktionen (FLMOPN, FLMGET, ...) und deren Parameter sind analog dem Handbuch von FLAM V4.x (Kapitel 3, Schnittstellen) angegeben, die Parameter sind wie im Handbuch nummeriert. Die n chste Angabe ist die Adresse des Parameters, der Parameterinhalt steht in Klammern '()'.

Die Kontrollbl cke ACB und RPL sind durch das Subsystem-Interface vorgegeben, auch wenn der Zugriff auf eine PS-Datei erfolgen sollte.

Zum Verst ndnis dieser Kontrollbl cke sei auf die Handb cher der IBM verwiesen, z.B. 'Macro Instructions for VSAM Data Sets', 'Data Areas' oder die Makros 'SYS1.AMODGEN(IFGACB)' UND 'SYS1.AMODGEN(IFGRPL)'.

10.8 Splitten der FLAMFILE

10.8.1 Serieller Splitt

Werden sehr gro e Dateien erzeugt oder sind maximale Dateigr en vorgegeben, so ist serieller Splitt der FLAMFILE m glich.

```
//AUSGABE   DD  DSN=dateiname1,
//              SUBSYS=(FLAM,'MO=ADC,SPLITM=SER,SPLITS=200')
```

Dateiname1 muss eine bereits katalogisierte Datei sein, der Name muss mindestens eine Ziffer enthalten. Bei Erreichen von 200 MB, wird *dateiname1* geschlossen und *dateiname2* erstellt. Diese und die folgenden Dateien m ssen nicht katalogisiert sein. Der ADC-Mode wurde hier nur wegen der besseren Komprimierung zus tzlich eingeschaltet.

Hei t die katalogisierte Datei z.B. FLAM.SUBDAT.ADC01, so w rden die Dateien FLAM.SUBDAT.ADC02, FLAM.SUBDAT.ADC03, usw. im Katalog gesucht oder ggf. erzeugt werden.

Sind die Dateinamen fest vorgegeben oder sollen flexiblere Allokationen gefordert sein, so empfiehlt sich die Verwendung der Zuordnung ber weitere DD-Statements.

```
//AUSGABE DD SUBSYS=(FLAM,'SPLITM=SER,SPLITS=200,FLAMDD=CMP01')
//*
//CMP01      DD  DSN=beliebig1,DISP=OLD
//CMP02      DD  DSN=beliebig2,DISP=OLD
//CMP03      DD  DSN=beliebig3,DISP=(NEW,CATLG),
//               UNIT=TAPE,…
```

Bitte beachten, dass alle Dateien die gleiche Satzl nge haben m ssen.

Zum Lesen einer seriell gesplitteten FLAMFILE sind keine Parameter notwendig:

```
//EINGABE   DD  DSN=dateiname1,
//              SUBSYS=FLAM
```

Weitere Dateien werden selbstt tig allokiert.

Die Zuordnung ber DD-Statements ist auch m glich:

```
//EINGABE   DD  SUBSYS=(FLAM,'FLAMDD=CMP01')
//*
//CMP01      DD  DSN=beliebig1,DISP=SHR
//CMP02      DD  DSN=beliebig2,DISP=SHR
//CMP03      DD  DSN=beliebig3,DISP=SHR
```

Die Reihenfolge der Dateien muss der bei der Erstellung entsprechen, d.h. CMP01 muss auch das erste Fragment der FLAMFILE zuordnen, usw.

10.8.2 Paralleler Splitt

Bei parallelem Splitt m ssen alle Dateien katalogisiert und in der JCL angegeben sein! Damit ist nur der Weg ber mehrere DD-Statements m glich.

Sind nicht alle Dateien zugeordnet, wird der FLAM-Fehlercode 531 protokolliert und das aufrufende Programm meldet einen OPEN-Fehler.

Sind die Dateien nicht alle katalogisiert (sondern werden z.B. mit DISP=NEW in der JCL angegeben), so kann es zum Systemfehler 50D f hren.

FLAM-sub bekommt den ersten DD-Namen als Parameter bergeben. Die weiteren werden selbstt tig gebildet.

```
//AUSGABE   DD  DSN=beliebig,
//       SUBSYS=(FLAM,'MO=ADC,SPLITM=PAR,SPLITN=3,FLAMDD=CMP01')
//*
//CMP01      DD  DSN=beliebig1,DISP=OLD
//CMP02      DD  DSN=beliebig2,DISP=OLD
//CMP03      DD  DSN=beliebig3,DISP=OLD
```

Bitte beachten, dass alle Dateien die gleiche Satzl nge haben m ssen. Es empfiehlt sich, alle Dateien mit gleichen Attributen anzulegen.

Analog zum Lesen:

```
//EINGABE   DD  DSN=beliebig,
//              SUBSYS=(FLAM,'FLAMDD=CMP01')
//*
//CMP01      DD  DSN=beliebig1,DISP=SHR
//CMP02      DD  DSN=beliebig2,DISP=SHR
//CMP03      DD  DSN=beliebig3,DISP=SHR
```

Die Reihenfolge der Dateien muss bei parallelem Splitt nicht identisch der Erstellung sein. FLAM-sub sortiert intern selbstt tig.

10.9 Verschl sselung

Zur Ver-/Entschl sselung w hrend des Lesens/Schreibens von Daten muss dem Subsystem ein Schl ssel bergeben werden.

Beim Schreiben muss der Verschl sselungsmodus (CRYPTOMODE) angegeben werden. Ohne Angabe wird FLAM angenommen (kompatibel zu FLAM-sub V3).

Beim Lesen ist das Verfahren in der FLAMFILE gespeichert und muss nicht extra angegeben werden.

10.9.1 **Verschl sselung mit Parameter CRYPTOKEY**

```
//DDNAME   DD  DSN=dateiname,
//             SUBSYS=(FLAM,'CRYPTOM=AES,CRYPTOK=OTTO')
```

Statt CRYPTOKEY kann auch PASSWORD angegeben werden (wie in FLAM-sub V3).

Bitte beachten, dass bei Schl sselangabe mit Hochkomma diese jeweils zu verdoppeln sind:

```
//DDNAME   DD  DSN=dateiname,
//             SUBSYS=(FLAM,'CRYPTOM=AES,CRYPTOK=X''AE01EA''')
```

10.9.2 Verschl sselung mit KMEXIT

Das Programm *name* bedient die KME-Schnittstelle und bergibt an das Subsystem einen Schl ssel. Dazu werden die Parameter *parameter* bergeben.

Es ist Angelegenheit dieses Programms, wie es an einen Schl ssel kommt (z.B. ber einen Anschluss an das HSM (Hoch Sicherheit Modul) im z/OS, ber die genormte KMI-Schnittstelle, ...).

```
//DDNAME   DD  DSN=dateiname,
//             SUBSYS=(FLAM,'CRYPTOM=AES',
//             'KME=name,KMP=C''parameter''')
```

10.9.3 Verschl sselung mit FKMEFILE

Das Programm FKMEFILE bedient die KME-Schnittstelle, liest aus einer Datei den Schl ssel und bergibt ihn an das Subsystem. Dazu werden der Parameter *ddname* bergeben.

Der erste Satz der Datei muss den Schl ssel enthalten. Der Schl ssel kann wie der Parameter CRYPTOKEY als String, in der Form C'string mit Leerzeichen', A'schl ssel' oder X'hexwerte' eingetragen sein. Leerzeichen nach dem Schl ssel werden im Datensatz ignoriert.

```
//DDNAME   DD  DSN=dateiname,
//             SUBSYS=(FLAM,'CRYPTOM=AES',
//             'KME=FKMEFILE,KMP=C''ddname''')
//* DIESE DATEI ENTHAELT DEN SCHLUESSEL
//ddname   DD  DSN=dateiname,DISP=SHR
```

Zum einfachen Test geht nat rlich auch

//*ddname* DD *
C'Das ist der Schl ssel'
/*

aber damit wird der Schl ssel im Job protokolliert, was ja genau vermieden werden soll.