



KfK 5029

Mai 1992

facile
Eine Sammlung von
Kommandoprozeduren
für den Alltag

K. Thurnay
Institut für Neutronenphysik und Reaktortechnik

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE
Institut für Neutronenphysik und Reaktortechnik

KfK 5029

facile
EINE SAMMLUNG VON KOMMANDOPROZEDUREN FÜR DEN ALLTAG

K. Thurnay

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript gedruckt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

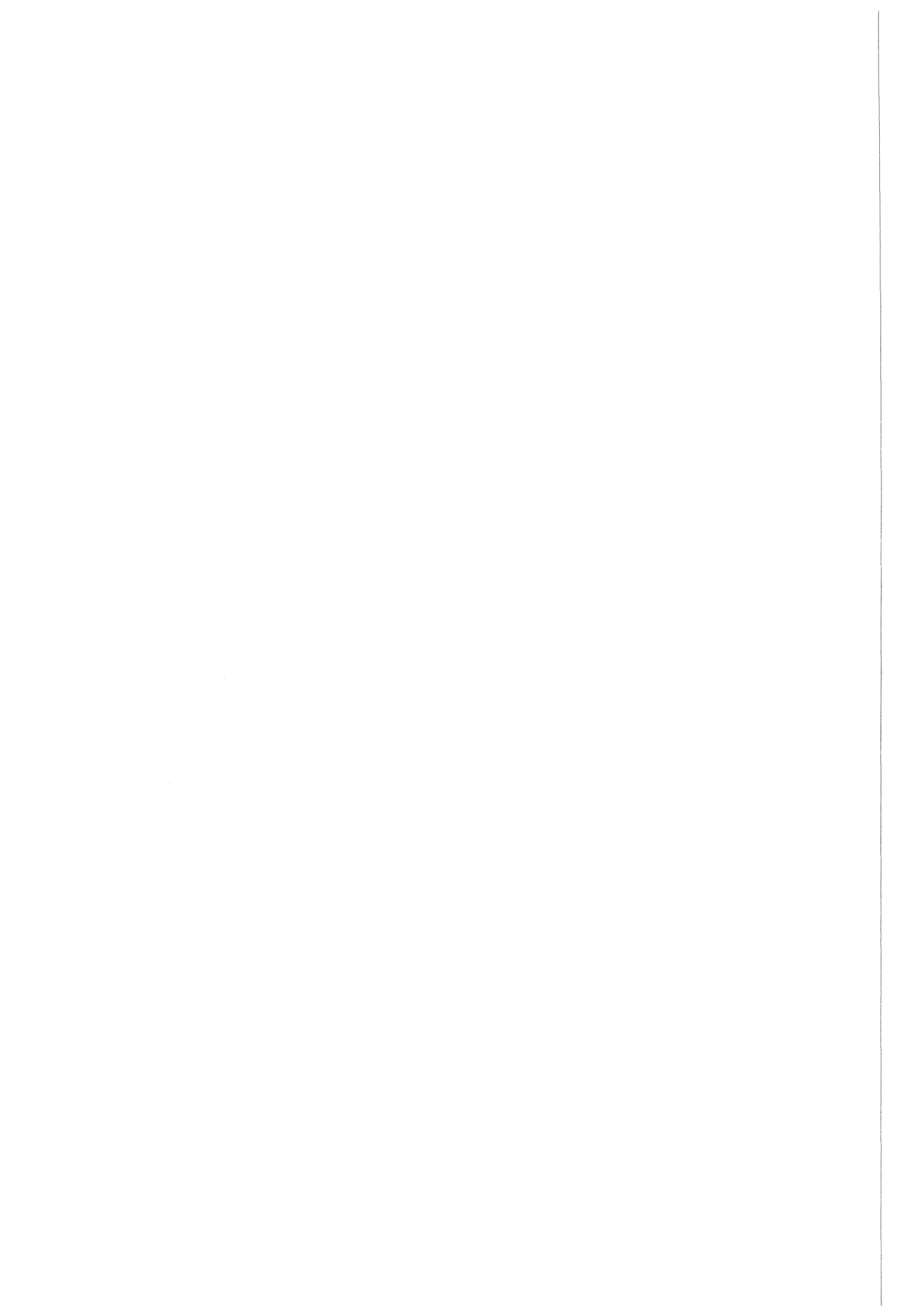
Zusammenfassung

facile ist eine Sammlung von Kommandoprozeduren, die die tägliche Arbeit am Bildschirm erleichtern. facile hilft bei der Verwaltung eigener Dateien, bei der Zusammenstellung von Rechenaufträgen, bei der Erzeugung von Bildern, beim Austesten von FORTRAN-Routinen usw.

Abstract.

facile. A set of Command Lists for the everyday use.

facile ist a set of CLISTs written to facilitate the regular use of TSO. facile helps in managing datasets, in constructing job-cards, in creating pictures, in testing FORTRAN-routines etc.



Kurzvorstellung der Prozeduren

ALFA	koppelt die eigenen Kommandoprozeduren zum System.
CORE	schiebt eine gegliederte Datei zusammen.
DAMA	räumt die Dateien des Benutzers auf.
DELCAT	löscht Katalogeinträge.
DRUCKT	gibt eine Datei des Benutzers durch den Schnelldrucker aus.
FORCO	unterstützt die FORTRAN-Kontrolle einer Datei im TSO.
FREI	befreit die eigene Kommandoprozedur vom System.
FUEGE	erleichtert die Handhabung des Kommandos MERGE.
GO	vereinfacht die Abgabe von Rechenaufträgen.
KATALOG	fügt die Namen der Glieder einer Datei zusammen.
OH	überträgt eine Outlist-Datei in eine andere Datei.
POST	hilft beim Transfer durch das EARN-Netz.
PROLET	erzeugt einen gebrauchsfertigen Load-Modul.
RACLI	erzeugt eine Liste der eigenen RACF-Zeilen.
SERIE	erzeugt aus einer gegliederten Datei eine serielle Datei.
SPY	erleichtert den Einstieg in SPEAKEASY.
SS	Handhabt die Dateien SPF1.LIST und SPFLOG3.LIST.
TAFEL	unterstützt den Einstieg in MAPPER.
TSOKO	lenkt die Ausgabe eines Kommandos in eine Datei um.
WEG	löscht eine Datei im Vordergrund oder im Massenspeicher
BRAUS	schickt GS-Bilder zu einem Zeichengerät.
IMMANI	verwaltet GS-Bilder.
PICOR	schreibt die Struktur eines GS-Bildes in eine Datei.
PIXI	führt durch eine GS-Sitzung.
PODEL	radiert und
PONUM	markiert in einem GS-Bild.
DATAS	kopiert SERVUS-Dateien im Vordergrund.
DDFIG	zeichnet eine Funktionsoberfläche.
DRUDA	druckt eine SERVUS-Datei aus.

ENDE	setzt einen neuen Endpunkt in eine SERVUS-Datei.
FIGUR	zeichnet eine Funktionsschar.
FUNKIN	erzeugt Funktionen im Vordergrund.
IDA	eröffnet eine neue SERVUS-Datei.
KOPIER	kopiert eine SERVUS-Datei.
LOESCH	radiert in eine SERVUS-Datei.
TEXIMA	macht aus Texten Bilder.

Inhaltsverzeichnis

Einführung	1
Hauptprozeduren	7
ALFA	7
CL	7
CORE	8
DAMA	8
DELCAT	10
DRUCKT	10
FORCO	12
FREI	13
FUEGE	13
GO	14
KATALOG	15
OH	16
POST	16
PROLET	17
RACLI	20
SERIE	20
SPY	22
SS	23
TAFEL	25
TSOKO	27
WEG	30
G.S.-Prozeduren	33
BRAUS	33
IMMANI	39
PICOR	42
PIXI	43
PODEL	44
PONUM	46
SERVUS-Prozeduren	49
DATAS	49
DDFIG	50
DRUDA	51
ENDE	53
FIGUR	54
FUNKIN	55
IDA	56
KOPIER	57
LOESCH	59
TEXIMA	61
Hilfsprozeduren	63
ADA	63
ALOGS	63
BEKI	65
DRUQQ	66
FEJ	67
GLIEDER	68
KRAM	69
NEVE	69

OBJEKT	70
SERVIN	72
TAGOK	74
TUKOR	74
Endprozeduren	77
HILDA	77
SUSA	77
TERI	78
Literatur:	79
Anhang A. Die Datei "Memory"	81
Anhang B. Wie man eine CLIST-Datei an das System anknüpft	83

Sinn und Zweck.

Kommandoprozeduren sind dazu da, das tägliche Leben des Bildschirm-Sklaven - zu deutsch TSO-USER - zu erleichtern. Man kann sie benutzen, um häufig auftretende Folgen von TSO-Kommandos automatisch ablaufen zu lassen, man kann sie aber auch verwenden, um Dateinamen, Prozedurnamen, Parameter u.s.w, die man sich nur schlecht merken kann, zu speichern (s. z. B. die Prozedur „FUEGE“ auf Seite 13 oder „SS“ auf Seite 23). Mit Hilfe von Kommandoprozeduren kann man, z. B. gegliederte Dateien in serielle umwandeln (KATALOG, SERIE) oder aus einer FORTRAN-Datei einen ausführbaren Load-modul bauen (PROLET). Man kann sogar Kommandoprozedur veranlassen, am Ende des Tages die Dateien des Benutzers aufzuräumen und konservieren (TERI). Eine ausführliche Beschreibung der Kommandoprozeduren gibt es in /1/.

Ziel dieser Arbeit war es, eine Anzahl von Kommandoprozeduren - die sich in langjährigem, täglichem Gebrauch als handlich erwiesen haben - auch anderen Benutzer zugänglich zu machen.

Anmerkung: zur Schreibweise : in diesem Bericht werden Angaben - die der Benutzer in einer Prozedur selber einsetzen/abändern muß - kleingeschrieben, die festgelegten Angaben erscheinen dagegen in Blockschrift. Zum Beispiel wird, bei inr000.MEMORY vom Benutzer erwartet, daß er hier seine eigene Benutzernummer, z.B. ILY111 einsetzt :

inr000.MEMORY ==> ILY111.MEMORY

Ausdrücklich möchte ich Herrn Dr. Dombrowsky vom INR danken, daß er seine Benutzernummer, inr000 als Beispiel freundlicherweise zur Verfügung gestellt hat.

Kommandoprozeduren

Kommandoprozeduren werden häufig auch als CLIST-s bezeichnet und ihr Dateiformat ist dementsprechend

bentzr.(name.) CLIST

Eine Kommandoprozedur besteht aus einer geordneten Reihe von TSO-Kommandos, aus Wert-Setzungen, aus logischen Verknüpfungen und aus Partikel, die zum Gespräch mit dem Benutzer und mit dem System dienen.

Folgendes Beispiel zeigt die Sehr Einfache Kommandoprozedur, "SEK" aus der Datei 'inr000.CLIST' :

ERASE DSMUTTOC.TEXT	00010001
ERASE GML.LIST	00020001
ERASE EML.LIST	00030001
ERASE EML.LISTING	00040001

Abbildung 1. Kommandoprozedur 'inr000.clist(SEK)'

Beim Aufruf

EXEC 'inr000.clist(SEK)'

beseitigt SEK die aufgeführte Dateien (die bei der GML-Textausgabe vorübergehend gebraucht wurden).

Gliederung.

Kommandoprozeduren können ihrerseits Kommandoprozeduren als TSO-Kommandos aufrufen. Diese Freiheit ermöglicht eine vernünftige Gliederung dieser Prozeduren. Häufig auftretende Teilaufgaben kann man in Unter- oder Hilfsprozeduren zusammenfassen und überprüfen. Die eigentlichen Hauptprozeduren benutzen dann diese Hilfsprozeduren und werden damit übersichtlicher und auch sicherer.

Prozedurparameter

Eine Kommandoprozedur kann - wie das Beispiel oben - in einer, für immer festgelegten Form ablaufen. Bei manchen Kommandoprozeduren kann aber der Benutzer in den Ablauf eingreifen, indem er der Kommandoprozedur Eingabedaten, Zahlen oder Zeichenketten mitgibt.

Die nächste Abbildung zeigt die Mäßig Einfache Kommandoprozedur, "MEK" als Beispiel für eine Kommandoprozedur mit Eingabedaten :

PROC 1 GLIED DATEI(TEST)	00010001
SET &N3 = &STR(&DATEI..CNTL)	00020001
SET &TON = &SYSUID..&N3(&GLIED)	00030001
SUBMIT '&TON'	00040001
EXIT	00050001

Abbildung 2. Kommandoprozedur 'inr000.clist(MEK)'

Der Prozedurkopf - die erste Zeile - von MEK beschreibt die möglichen Eingaben bei dieser Prozedur. Es wird 1 Positionsparameter (ein Name) an der Stelle GLIED erwartet und außerdem der Schlüsselwortparameter DATEI. DATEI ist zunächst mit dem Name "TEST" vorbesetzt.

Anmerkung: Bei mehreren Parametern ist die Reihenfolge der Schlüsselwortparameter beliebig, aber die Positionsparameter müssen auf jeden Fall als die ersten und in der angegebenen Reihenfolge kommen.

Beim Vergessen des Positionsparameters wird der Benutzer vom System ermahnt :

ENTER POSITIONAL PARAMETER GLIED -

Beim Weglassen des Schlüsselwortparameters DATEI benutzt MEK die Vorbesetzung TEST.

Falls der Benutzer beim Aufruf die Eigenschaften "GLIED" und "DATEI" mit geeigneten Eingaben (MONOF bzw. AQUA) aufgefüllt hatte

```
EXEC 'inr000.CLIST(MEK)' 'MONOF DATEI(AQUA)'
```

wird das Dateiglied 'inr000.AQUA.CNTL(MONOF)' als Rechenauftrag abgeschickt ("subgemittet").

Aufruf

Eine Kommandoprozedur kann man, wenn man penibel ist, wie oben, mit dem vollen Namen ansprechen. Es geht aber auch salopper

```
EX (MEK) 'MONOF DATEI(AQUA)'
```

denn - da die Hochkommata am Dateinamen fehlen - nimmt TSO an, daß es sich um eine namenlose CLIST-Datei handelt und daß der Name mit der Benutzernummer 'inr000' beginnt.

Man kann schließlich - wenn man sowieso in SPF-Umgebung arbeitet - im AHF-Menü die Prozedurdatei 'inr000.CLIST' fest an die System-Prozeduren andocken (S. Anhang B, „Wie man eine CLIST-Datei an das System anknüpft“ auf Seite 83). In diesem Falle vereinfacht sich der Aufruf zu

```
MEK MONOF DATEI(AQUA)
```

Wo kann man den Aufruf eingeben ?

Am einfachsten ist es, man benutzt das TSO Menu (# 6) des ISPF :

```
----- TSO COMMAND PROCESSOR -----  
ISPF COMMAND ==>  
  
ENTER TSO COMMAND, CLIST, OR REXX EXEC BELOW:  
==> MEK MONOF DATEI(AQUA)
```

Man kann aber auch z. B. im EDIT-Menü (ISPF # 2) ein Kommandoprozedur ablaufen lassen, wenn man das dortige COMMAND-Feld wie folgt beschreibt :

```
EDIT ---- INR105.CLIST.TEXT(EINFUH) - 01.09 ----- COLUMNS 001 072
----- EDIT - ENTRY PANEL -----
COMMAND ==> TSO MEK MONOF DATEI(AQUA)
```

ISPF LIBRARY:

```
PROJECT ==> INR105
GROUP   ==> CLIST      ==>          ==>          ==>
TYPE    ==> TEXT
MEMBER  ==>                (Blank or pattern for member selection list)
```

OTHER PARTITIONED OR SEQUENTIAL DATA SET:

```
DATA SET NAME ==>
VOLUME SERIAL ==>        (If not cataloged)

DATA SET PASSWORD ==>        (If password protected)

PROFILE NAME     ==>        (Blank defaults to data set type)

INITIAL MACRO    ==>        LMF LOCK ==> YES    (YES, NO or NEVER)

FORMAT NAME      ==>        MIXED MODE ==> NO    (YES or NO)
```

Wenn man MEK dagegen im TSO-Grundzustand aufrufen will, muß man sie - da die CLIST-Datei jetzt noch nicht/nichtmehr an der System-Prozeduren angeknüpft ist - mit dem Kommando EXEC ansprechen :

```
READY
EX (MEK) 'MONOF DATEI(AQUA)'
```

Hinweise.

- Beim Kommando-Dialog sollte man die folgenden Zeichen nicht am Bildschirm eingeben :

```
+ - * / = > < | ~ &  
( , '
```

Die ersten 10 dieser Zeichen führen zum sofortigen Abbruch der Prozedur, die letzten 3 werden als Leerzeichen mißdeutet.

- Bei Kommandoprozeduren - die auch FORTRAN-Prozeduren benutzen (PROLET, SERVUS-) - achten Sie bitte darauf, daß beim Einschalten

```
----- TSO/E LOGON -----  
PF1/PF13 ==> Help   PF3/PF15 ==> Logoff   PA1 ==> Attention   PA2 ==> Reshow  
You may request specific HELP information by entering a '?' in any entry field.  
ENTER LOGON PARAMETERS BELOW:                RACF LOGON PARAMETERS:  
  
USERID    ==> inr000                            NEW PASSWORD ==>  
PASSWORD  ==>                                    GROUP IDENT  ==>  
  
PROCEDURE ==> FV  
  
ACCT NMBR ==> 000-999-abcde  
SIZE      ==> 6000  
PERFORM   ==>  
COMMAND   ==>  
  
ENTER AN 'S' BEFORE EACH OPTION DESIRED BELOW:  
  
-NOMAIL      -NONOTICE      -RECONNECT      -OIDCARD
```

PROCEDURE F oder FV gesetzt ist. Sonst kann es vorkommen, daß die Kanäle FT15 und FT16 nicht angeschlossen sind und die FORTRAN Ein- und Ausgabe nicht klappt !

- Bei mancher Prozedur wird eine Hilfsdatei neu angelegt und bearbeitet. Dies erzeugt am Bildschirm die Nachricht :

```
INPUT  
00010 -
```

Einfach auf die "ENTER"-Taste drücken.

- Es kommt immer wieder vor, daß man sich vertippt und die Kommandoprozedur sich verheddert. Mit der Taste **Attn** kann man sich im solchen Falle befreien.

Wie kann man sich die facile-Prozeduren beschaffen ?

- Die aufgeführten Prozeduren sind in der Datei

```
'INR105.CLIST'
```

enthalten.

Der Benutzer wird gebeten, eine eigene Datei 'inr000.CLIST' anzulegen und die einzelnen Glieder zu kopieren. Dieses Vorgehen erlaubt es ihm, die Prozeduren entsprechend den eigenen Bedürfnissen zurechtzuschneiden. Zu verändern braucht er nichts, bis auf eine Ausnahme : in der zweiten Zeile der Prozedur **FEJ**

```
USER(dombrowsky) ACCOUNT('0000,999,abcde')
```

muß er die entsprechenden eigenen Benutzer-Angaben einsetzen. Dasselbe gilt für die zweite Zeile von hinten dieser Prozedur :

```
IF &MA = &STR(0) THEN 0111 /*FORMAT PR,DDNAME=,DEST=Pinr1
```

wo der Benutzer den Namen des lokalen Druckers ersetzen soll.

- Weiterhin wird noch die Hilfsdatei

```
'inr000.MEMORY'
```

gebraucht.

Diese Datei (s. Anhang A, „Die Datei "Memory"“ auf Seite 81) dient dazu, die Eingaben, die die Prozeduren zuletzt benutzt hatten zu speichern und als nächste Vorbe- setzung bereitzuhalten. Man kann sie sich beschaffen, indem man die Datei

```
'INR105.MEMORY'
```

kopiert.

- Man braucht schließlich noch häufig (SPY, TAFEL, ALOGS) den graphischen Speicher

```
'inr000.ADMGDF'
```

Durch einmaligen Aufruf der GDDM-Menu GDDMICU (# 1) wird diese Datei automatisch angelegt.

Im Folgendem werden die facile-Prozeduren kurz beschrieben.

- Letzte Warnung! Die Benutzung der nicht dargestellten Prozeduren kann für Ihre gute Laune abträglich sein !

ALFA

ALFA verknüpft die eigenen Kommandoprozeduren und Edit-Macros - hier als inr000.CLIST resp. inr000.MACRO.CLIST geführt - mit den entsprechenden Systemprozeduren.

Die so zugefügten Kommandoprozeduren kann man dann im Folgendem schon allein mit dem Member-Name Aufrufen :

anstatt EX (ALFA)
genügt ALFA

ALFA benötigt eine Hilfsprozedur (s. „Hilfsprozeduren“)

- inr000.CLIST(GLIEDER)

Die Prozedurliste:

```
PROC 0
/* INITIATE                               92/03/09
CONTROL PROMPT MAIN MSG
SET &TON = &SYSUID..CLIST
IF &SYSDSN('&TON') = OK THEN GOTO FIN
SET &DAN = '&TON'
SET &WON = &SYSUID..MACRO.CLIST
IF &SYSDSN('&WON') = OK THEN SET &DAN = &DAN '&WON'
WRITE
ALLOC F(SYSPROC) DA( 'TSOSYS.CMDPROC.CLIST' +
                    'TSOSYS.ISRCLIB' 'TSOSYS.ADICLIB' &DAN ) SHR REU
%GLIEDER &TON
IF &SYSDSN('&WON') = OK THEN %GLIEDER &WON
FIN: EXIT
```

CL

Ein Beispiel für eine Kommandoprozedur, die von der Kommandoprozedur PROLET angelegt wurde:

```
ALLOC DA('inr000.AQUA.DATA') F(FT15F001) SHR REU
ALLOC DA('inr000.AQUA.DATA') F(FT16F001) SHR REU
CALL 'inr000.AQUA.LOAD($TWSSR)'
FREE DA('inr000.AQUA.DATA')
FREE DA('inr000.AQUA.DATA')
```

CORE

CORE schiebt eine gegliederte Datei zusammen, und gibt den dabei gewonnenen Raum frei.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)

Benötigte Hilfsdatei :

- inr000.MEMORY(CORE)

Die Prozedurliste:

```
PROC 0
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE EINE DATEI WIRD ZUSAMMENGESCHOBEN          91/02/20
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(CORE)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &N3
SET &KAMEL=&STR()
EX '&SYSUID..CLIST(NEVE)'
IF &TON = &STR() THEN GOTO FIN
IF &M1 = &STR(#) THEN GOTO FIR
EX '&SYSUID..CLIST(BEKI)' 'MEM(CORE)'
COMPRESS '&TON' NOLIST
FIR: REDUCE '&TON'
SPACE '&TON'
FIN: EXIT
```

DAMA

DAMA ist eine Dateipflegerin. Sie komprimiert und massenspeichert Dateien, deren Namen in der Datei 'inr000.MEMORY(DAMA)' vor der Zeile ENDE aufgelistet sind. Ein Beispiel für diese Hilfsdatei :

```
inr000.blaues.kreuz
inr000.gruen.kohl
inr000.iecch.TEXT
inr000.rosen.kohl
inr000.rosen.montag
inr000.rot.kohl
inr000.KFK
ENDE
inr000.gruenes.kreuz
inr000.popanz.DATA
inr000.otes.kreuz
inr000.wirsing.kohl
inr000.CLIST
```

Abbildung 3. Die Hilfsdatei "DAMA"

Im obigen Beispiel werden die Dateien von 'inr000.blaues.kreuz' bis 'inr000.KFK' bearbeitet.

Anmerkung: Man sollte darauf achten, daß diese Datei keine Zahlen in den Kolonnen 73-80 aufweist!

DAMA benötigt neben der Hilfsdatei

- inr000.MEMORY(DAMA)

auch die Hilfsprozedur

- inr000.CLIST(KRAM)

Die Prozedurliste:

```
PROC 0 IND(YYY)
CONTROL PROMPT MAIN MSG END(QUIT)
ALLOC F(SYSPROC) DA( 'TSOSYS.CMDPROC.CLIST' +
    'TSOSYS.ISRCLIB' 'TSOSYS.ADICLIB' ) SHR REU
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE PROZEDUR DAMA
WRITE DATEIEN, DEREN NAMEN IN MEMORY(DAMA) STEHEN
WRITE WERDEN - GNADENLOS - MANIPULIERT
WRITE
SET &QUELLE=&SYSUID..MEMORY(DAMA)
ALLOC FILE(INTEX) DA('&QUELLE') SHR REU
OPENFILE INTEX INPUT
SET M = 0
DO WHILE &M <= 200
    GETFILE INTEX
    SET &SYSDVAL=&INTEX
    SET &TYN = &SYSDVAL
    IF &TYN = &STR(ENDE) THEN GOTO I5
    SET &P = &SUBSTR(1:1,&TYN)
    IF &P = &STR(I) THEN SET &P = &STR(BS)
    ELSE SET &P = &STR(TS)
    SET &TYN = &STR('&TYN')
    SET &M = &M+1
    WRITE &M.. &TYN
    IF &SYSDSN(&TYN)=OK THEN +
        DO
            COMPRESS &TYN NOLIST
            REDUCE &TYN
            SPACE &TYN
            &P &TYN
        QUIT
    ELSE WRITE &TYN IS NOT ACCESIBLE
    QUIT
I5: CLOSFIL INTEX
ERROR OFF
WRITE
FREE FILE(INTEX)
ALLOC F(SYSPROC) DA( 'TSOSYS.CMDPROC.CLIST' +
    'TSOSYS.ISRCLIB' 'TSOSYS.ADICLIB' +
    '&SYSUID..CLIST' +
    '&SYSUID..MACRO.CLIST') SHR REU
EX (KRAM) 'IND(&IND)'
FIN: EXIT
```

91/09/20

DELCAT

Mit DELCAT kann man Katalogeinträge während einer TSO-Sitzung löschen. Die Prozedur dient eigentlich nur dazu, die wichtige Stichwort "NOSCRATCH" des Kommandos DELETE zu behalten.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)

Benötigte Hilfsdatei :

- inr000.MEMORY(DELCAT)

Die Prozedurliste:

```
PROC 0
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE EIN KATALOG-EINTRAG WIRD GELOESCHT           91/02/21
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(DELCAT)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &N3
SET &KAMEL=&STR()
EX '&SYSUID..CLIST(NEVE)'
IF &TON = &STR() THEN GOTO FIN
EX '&SYSUID..CLIST(BEKI)' 'MEM(DELCAT)'
WRITE LOESCHEN ? ( #=JA | SONST NEIN )
READ &ANS
IF &ANS = &STR(#) THEN DELETE '&TON.' NOSCRATCH
FIN: EXIT
```

DRUCKT

Die Prozedur stellt einen IEBTPCH-Auftrag zusammen - um eine Datei oder Glieder einer Datei am Schnelldrucker auszugeben - und schickt ihn ab.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(DRUQQ)
- inr000.CLIST(FEJ)
- inr000.CLIST(GLIEDER)
- inr000.CLIST(NEVE)
- inr000.CLIST(SUSA)
- inr000.CLIST(TAGOK)

Benötigte Hilfsdateien :

- inr000.MEMORY(DRUCKT)
- inr000.MEMORY(FEJ)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE PROZEDUR      DRUCKT                      91/02/21
WRITE EINE DATEI WIRD | GLIEDER EINER DATEI WERDEN AUSGEDRUCKT
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(DRUCKT)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &N3 +
&MB1 &MB2 &MB3 &MB4 &MB5 &MB6 &MB7 &MB8 &MB9 &MB10 &MB11
SET &KAMEL=&STR()
WRITE QUELLDATEI :
SET &M1=&MB1
EX '&SYSUID..CLIST(NEVE)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &I = 1
IF &M1=&STR(#) THEN DO
    SET &OQ=&STR(PS)
    GOTO LNS
    QUIT
SET &OQ=&STR(PO)
WRITENR ALLE GLIEDER AUSDRUCKEN =: ( "ENTER" = JA )
READ &ANS
IF &ANS = &STR() THEN GOTO LNS
DO WHILE &I <= 20
    SET V = &&MB&I
    SET &M1 = &V
    EX '&SYSUID..CLIST(TAGOK)'
    IF &M1 = &STR(!) THEN GOTO LNS
    SET &KAMEL = &KAMEL &M1
    SET &MB&I = &M1
    SET &I = &I+1
    QUIT
LNS: SET &I = &I-1
EX '&SYSUID..CLIST(BEKI)' 'MEM(DRUCKT)'
EX '&SYSUID..CLIST(FEJ)'
EX '&SYSUID..CLIST(DRUQQ)'
EDIT '&SYSUID..&N3..CNTL' OLD CNTL NUM
270 PRINT TYPORG=&OQ,MAXFLDS=1&M1
IF &I <= 0 THEN 271 RECORD FIELD=(&N2)
ELSE DO
    SET &J = 270
    C &J =1=&I,MAXNAME=&I
    SET &K = 1
    DO WHILE &K <= &I
        SET &J = &J+1
        SET &V = &&MB&K
        &J MEMBER NAME=&V
        SET &J = &J+1
        &J RECORD FIELD=(&N2)
        SET &K = &K+1
    QUIT
    QUIT
LIST 100 400
EX '&SYSUID..CLIST(SUSA)' '&N3.'
FIN: EXIT

```

FORCO

FORCO unterstützt die FORTRAN77-Kontrolle einer FORT-Datei mit Hilfe des Kommandos FORT77 in einer TSO-Sitzung. Seit einiger Zeit kann man diese Aufgabe auch mit dem EDIT-Kommando "CHECK LN(F7)" erledigen. CHECK hat allerdings Nachteile gegenüber FORCO : z.B. benutzte aber nicht definierte Größen findet er nicht.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(GLIEDER)
- inr000.CLIST(NEVE)
- inr000.CLIST(TAGOK)

Benötigte Hilfsdatei :

- inr000.MEMORY(FORCO)

Die Prozedurliste:

```
PROC 0
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE FORDERGRUND-UEBERSETZEN ( FORTRAN77 )          91/02/25
WRITE
SET &EA=0
EX (BEKI) 'MEM(FORCO)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &M1 &MA &SO
SET &KAMEL=&STR()
SET &N3=&STR(FORT)
EX (NEVE) 'LQ(1)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &OPE = &STR()
IF &M1 = &STR(#) THEN DO
  EX (TAGOK)
  IF &M1 = &STR(!) THEN GOTO FIN
  SET &OPE = &STR(ELM(&M1))
END
WRITENR MAP ? / &MA / =: ( "ENTER"(=JA) | MAP | NOMAP )
READ &ANS
IF &ANS = &STR() THEN SET &MA = &ANS
WRITENR SOURCE ? / &SO / =: ( "ENTER"(=JA) | SOU | NOSOU )
READ &ANS
IF &ANS = &STR() THEN SET &SO = &ANS
SET &KAMEL = &KAMEL &M1 &MA &SO
EX (BEKI) 'MEM(FORCO)'
WRITE UEBERSETZUNG :
FORT77 '&TON' &OPE TERM NOGO NOOBJECT +
LANGLVL(77) &SO STATIS.LMSG PRINT(*) NOXREF &MA
FIN: EXIT
```

FREI

Die Prozedur befreit die eigene Kommandoprozedur aus der - mit der Kommando-
prozedur ALFA erzeugten - Verkettung.

Die Prozedurliste:

```
PROC 0
CONTROL PROMPT MAIN MSG
WRITE EIGENE PROZEDUREN BEFREIEN                               91/02/25
ALLOC F(SYSPROC) DA( 'TSOSYS.CMDPROC.CLIST' +
                    'TSOSYS.ISRCLIB' 'TSOSYS.ADICLIB' ) SHR REU
EXIT
```

FUEGE

Diese Prozedur erleichtert die Handhabung des TSO-Kommandos MERGE, indem sie
alle notwendigen Daten vom Benutzer erfragt.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(GLIEDER)
- inr000.CLIST(NEVE)
- inr000.CLIST(TAGOK)

Benötigte Hilfsdatei :

- inr000.MEMORY(FUEGE)

Die Prozedurliste:

```
PROC 0 VON(1) BIS(99999)
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE PROZEDUR FUEGE                                         91/02/25
WRITE EINE DATEI / EIN TEIL EINER DATEI WIRD
WRITE IN EINE ANDERE DATEI EINGEFUEGT
SET &EA=0
EX (BEKI) 'MEM(FUEGE)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &N3 &M1 &PROF &NAMEF &TYPF &GLIEDF
SET &KAMEL = &STR()
WRITE ZU KOPIEREN IST :
EX (NEVE)
IF &N1 = &STR(#) THEN GOTO FIN
SET &DATEIM=&TON
IF &M1=&STR(#) THEN DO
    EX (TAGOK)
    SET &DATEIM=&STR(&DATEIM.(&M1))
END
SET &KAMEL = &KAMEL &M1
QUP: IF &BIS=&STR() THEN WRITE KOPIERT WIRD NUR DIE &VON . ZEILE
ELSE WRITE KOPIERET WIRD VON DER &VON . BIS ZUR &BIS . ZEILE
WRITENR GUELTIGE ANTWORTEN : "ENTER"=JA | VON | BIS | ALL
READ &ANS
IF &ANS=&STR() THEN GOTO QUE
IF &ANS=&STR(VON) THEN DO
```

```

WRITE VON / &VON / =:
READ &VON
GOTO QUP
END
IF &ANS=&STR(BIS) THEN DO
  WRITENR BIS / &BIS / =: ( "ENTER" = NUR DIE &VON . ZEILE )
  READ &BIS
  GOTO QUP
  END
IF &ANS=&STR(ALL) THEN SET &VON=&STR()
QUE: IF &VON = &STR() THEN SET &BIS = &STR()
IF &BIS=&STR() THEN SET &UM=&STR(&STR(&VON)-&STR(&BIS))
ELSE SET &UM=&STR(&VON)
ZIEL: WRITE KOPIERET WIRD IN DIE DATEI :
SET &N1=&PROF
SET &N2=&NAMEF
SET &N3=&TYPF
SET &M1=&GLIEDF
EX (NEVE)
IF &N1 = &STR(#) THEN GOTO FIN
SET &DATEIF=&TON
IF &M1=&STR(#) THEN DO
  EX (TAGOK)
  SET &DATEIF=&STR(&DATEIF.(&M1.))
  END
SET &KAMEL = &KAMEL &M1
WO: WRITE DIE EINFUEGUNG BEGINNT BEI DER ZEILE =:
WRITENR ( E = NACH DER LETZTEN | 0 = VOR DER ERSTEN ZEILE )
READ &AT
IF &AT = &STR(E) THEN SET &AT=&STR()
EX (BEKI) 'MEM(FUEGE)'
IF &AT=&STR() THEN SET &UM=&STR(&UM&STR( BEI &AT IN))
ELSE SET &UM=&STR(&UM&STR( AM ENDE VON))
MERGE '&DATEIM' &VON &BIS '&DATEIF' &AT NONUM NONUM2
IF &LASTCC = 0 THEN WRITE '&DATEIM' &UM '&DATEIF' EINGEFUEGT.
FIN: EXIT

```

GO

GO vereinfacht die Abgabe von häufig vorkommender Rechenaufträge. Besonders vorteilhaft ist GO, wenn man andauernd mit ein-und-derselben CNTL-Datei arbeitet. Man kann dann den Auftrag aus dem COMMAND-Feld jeder SPF-Menu als "TSO GO glied" abschicken. Tippfehler werden geduldet.

GO benötigt weder Hilfsprozeduren noch Hilfsdateien.

NAME ist der Name des abzuschickenden Gliedes in der CNTL-Datei **CAD**.

Beispiel : GO B1 CAD(servus)

Die Prozedurliste:

```

PROC 1 NAME CAD(MUMUS)
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE JOBABGABE=SCHNELLSCHUSS
SET &N3 = &STR(&CAD..CNTL)
SET &TON = &SYSUID..&N3
SET &M1 = &NAME

```

92/03/09


```

EX (TAGOK)
IF &M1 = &STR(!) THEN GOTO FIN
SET &TON = &TON.(&M1.)
SUBMIT '&TON'
FIN: EXIT

```

KATALOG

Diese Prozedur fügt die vollen Namen sämtlichen Glieder einer Datei zu einem Katalog zusammen. Der Katalog wird in die Datei 'inr000.MEMO' abgelegt.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)

Benötigte Hilfsdatei :

- inr000.MEMORY(KATALOG)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG END(STOP)
GLOBAL KAMEL EA N1 N2 N3 M1 TON
SET &DAK = &STR(&SYSUID..MEMO)
WRITE PROZEDUR KATALOG
WRITE DIE VOLLSTAENDIGEN NAMEN DER GLIEDER EINER DATEI
WRITE WERDEN IN DIE DATEI &DAK ZUSAMMENGETRAGEN.
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(KATALOG)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &N3
SET &KAMEL=&STR()
WRITE ANGABEN ZU DER QUELLDATEI :
EX '&SYSUID..CLIST(NEVE)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &QUELLE=&TON
EX '&SYSUID..CLIST(BEKI)' 'MEM(KATALOG)'
WRITE
WRITE KATALOGHERSTELLUNG:
WRITE
ERASE '&DAK'
EX '&SYSUID..CLIST(ADA)' '&DAK LIN(80)'
ALLOC FILE(INTEX) DA('&DAK') SHR REU
OPENFILE INTEX OUTPUT
SET &SYSOUTTRAP = 1000
LISTDS '&QUELLE' MEMBERS
SET I = 7
SET M = 0
ERROR GOTO I2
DO WHILE &I <= &SYSOUTLINE
  IF &I > &SYSOUTTRAP THEN GOTO EL
  SET DSN = &&SYSOUTLINE&I
  SET MEM = &DSN
  SET DAT = &QUELLE.(&MEM)
  SET INTEX = &DAT

```

92/02/28

```

PUTFILE INTEX
SET &M = &M+1
WRITE &M.. &DAT
I2: SET &I = &I+1
STOP
EL: CLOSFIL INTEX
ERROR OFF
WRITE
WRITE &M GLIEDER AUS &QUELLE IM KATALOG '&DAK'
FIN: EXIT

```

OH

OH überträgt eine Outlist-Edit-Datei in eine andere Datei des Benutzers. Weder Hilfsprozeduren, noch Hilfsdateien werden benötigt.

Anmerkung: Durch die Beschaffung der "SPOOL DISPLAY FACILITY (SDF)", /2/ durch HDI ist diese Prozedur überflüssig geworden.

Die Prozedurliste:

```

PROC 0 N1(&SYSUID.) N3(TEXT) M1(LO) NU(111)
CONTROL PROMPT MAIN MSG
WRITE PROZEDUR OH 91/09/25
WRITE EINE OUTLIST-EDIT-DATEI WIRD KOPIERT UND GELOESCHT.
WRITE
L1: WRITENR SPF=NUMMER DER DATEI / &NU / =: ( 0 = "ENDE" )
READ &ANS
IF &ANS=0 THEN GOTO FIN
IF &ANS=&STR() THEN SET &NU=&ANS
SET &NAM = &STR(SPF&NU)
SET &TON = &STR(&SYSUID..&NAM..OUTLIST)
IF &SYSDSN('&TON')=&OK THEN DO
WRITE &TON &SYSDSN('&TON') !
WRITE
WRITE KATALOG DER &N1 -DATEIEN :
WRITE
CAT &N1
GOTO L1
END
SET &ZIEL = &STR(&N1..&N3.(&M1))
COPY '&TON' '&ZIEL' NONUM
ERASE '&TON.'
WRITE DATEI '&TON' IN DIE DATEI '&ZIEL' KOPIERT UND GELOESCHT.
FIN: EXIT

```

POST

Diese Prozedur unterstützt die Datei-Übertragung durch das EARN-Netz. Dient eigentlich auch nur als Memo für unaussprechliche Namen.

Benötigte Hilfsprozedur :

- inr000.CLIST(NEVE)

Die Prozedurliste:

```

PROC 0 WO(WISCVM) WER(SMTPUSER)
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE PROZEDUR POST
WRITE EINE DATEI WIRD DURCH DEN EARN-NETZ GESCHICKT.
WRITE
SET &N2=&STR(POST)
SET &N3=&STR(DATA)
EX (NEVE)
IF &N1 = &STR(#) | &M1 == &STR(#) THEN GOTO FIN
WRITENR WOHN / &WO / ( ENTER=JA | WISCVM | LOCAL ) =:
READ &ANS
IF &ANS == &STR() THEN DO
  IF &SUBSTR(1:1,&ANS)=&STR(L) THEN SET &ANS=&STR(LOCAL)
  SET &WO = &ANS
END
WRITENR AN WEM / &WER / =: ( "ENTER"=JA | SMTPUSER | IRE748 | ... )
READ &ANS
IF &ANS == &STR() THEN SET &WER = &ANS
WRITE BEGINN DER PROZEDUR
TRANSMIT &WO..&WER. DSNAME('&TON')
IF &LASTCC = 0 THEN WRITE DATEI &TON NACH &WO..&WER ABGESCHICKT .
FIN: EXIT

```

91/02/26

PROLET

PROLET erzeugt anhand einer FORT-Datei einen gebrauchsfertigen Load-Modul. Zum Modul kan man die

- TRACEGS,
- KATHER und
- SERVUS

Prozeduren zulinken und zwei Hilfsdateien zufügen.

Abschließend erzeugt PROLET die Kommandoprozedur inr000.CLIST(CL) (s. Beispiel „CL“ auf Seite 7), die den Aufruf des Moduls enthält. Man kann anschließend den fertigen Modul wie folgt aktivieren :

EX (CL)

Sehr nützliche Prozedur, wenn man eine FORTRAN-Prozedur im Rahmen einer TSO-Sitzung mit wechselnder Eingaben auf Herz und Nieren prüfen will. Wie man eine normale FORTRAN-Prozedur umstellt, damit sie am Bildschirm Daten empfangen und wiedergeben kann, daß kann man z. B. in /3/ nachlesen.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(GLIEDER)
- inr000.CLIST(NEVE)
- inr000.CLIST(TAGOK)

Benötigte Hilfsdatei :

- inr000.MEMORY(PROLET)

Die Prozedurliste:

```

PROC 0 PROG(TEMP) DEB(' ') HEL(' ')
CONTROL PROMPT MAIN MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 M1 TON
SET &NOTA=&STR(MEMBER SPECIFIED, BUT DATASET IS NOT PARTITIONED)
SET &SER = &STR(INR105.SERVUS.LOAD)
WRITE PROZEDUR          PROLET                               92/02/07
WRITE ERSTELLUNG EINES LOAD - MODULS ZU EINER DIALOG-PROZEDUR
WRITE
SET &EA=0
EX (BEKI) 'MEM(PROLET)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &M1 &PROL &NAML +
&CONC &D1 &PRO1 &NAM1 &TYP1 &D2 &PRO2 &NAM2 &TYP2
SET &KAMEL=&STR()
WRITE QUELL-("FORT")-DATEI
SET &N3=&STR(FORT)
WIE: EX (NEVE) 'LQ(1)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &DATF=&TON
IF &M1 = &STR(#) THEN DO
    EX (TAGOK)
    IF &M1 = &STR(!) THEN GOTO WIE
    SET &DATF=&DATF(&M1)
    SET &GLIED=&M1
    QUIT
SET &KAMEL=&KAMEL &M1
WRITE DEBUG ? ( "ENTER"=NEIN ) =:
READ &ANS
IF &ANS=&STR() THEN DO
    SET &DEB=&STR(DEBUG(INIT,TRACE,ARGCHK,SUBTRACE,UNDEF))
    SET &HEL=&STR(MAP LIST)
    QUIT
WRITE ZIEL-("LOAD")-DATEI
SET &N1=&PROL
SET &N2=&NAML
SET &N3=&STR(LOAD)
EX (NEVE) 'LQ(1) CQ(1)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &DATL=&TON
UMG: WRITE BENOETIGTE HILFSMODULE VOM SYSTEM / &CONC / =:
WRITENR ( "#"=KEINE | TRACEGS (TR) | KATHER (KT) | +
ALLES ANDERE = NUR SERVUS )
READ &ANS
IF &ANS = &STR() THEN SET &CONC=&ANS
SET &KAMEL = &KAMEL &CONC
WRITENR WIRD EIN CALL-PROZEDUR BENOETIGT ? ( ENTER = NEIN ) =:
READ &ICA
IF &ICA=&STR() THEN GOTO IPR
SET &M1=&STR(#)
SET &I=0
DO WHILE &I < 2
    DE: SET &I=&I+1
    SET D=&&D&I
    SET P=&&PRO&I
    SET Q=&&NAM&I
    SET T=&&TYP&I
    WRITE NR. DER &I.. HILFSDATEI / &D / ( 0 = KEINE ) =:
    READ &ANS
    IF &ANS = 0 THEN DO

```

```

        SET &I=&I-1
        GOTO BEG
        QUIT
    IF &ANS=&STR() THEN SET &D=&ANS
    SET &KAMEL = &KAMEL &D
    SET &N1=&P
    SET &N2=&Q
    SET &N3=&T
    EX (NEVE)
    IF &N1 = &STR(#) THEN GOTO BEG
    IF &M1 = &STR(#) THEN DO
        EX (TAGOK)
        IF &M1 = &STR(!) THEN GOTO BEG
        SET &TON=&STR(&TON(&M1))
        QUIT
    IF &LENGTH(&STR(&D)) < 2 THEN SET &D=&STR(0&D)
    SET &F&I=&STR(FT)&STR(&D)&STR(F001)
    SET &DAT&I=&TON
    QUIT
BEG: IF &I < 1 THEN SET &KAMEL=&KAMEL &D1 &PR01 &NAM1
    IF &I < 2 THEN SET &KAMEL=&KAMEL &D2 &PR02 &NAM2
    IPR: EX (BEKI) 'MEM(PROLET)'
    WRITE BEGINN DER PROZEDUR
    SET &COZ = &STR()
    SET &COS = &STR('SYS7.FORTLIB' 'SYS2.FORTLIB' +
        'SYS2.GS7' )
    IF &SYSDSN('&DATL.(&GLIED)')=&NOTA THEN ERASE '&DATL'
    IF &SYSDSN('&DATL')=&OK THEN +
        EX (ADA) '&DATL LIN(0) KA(5)'
    IF &CONC = &STR(#) THEN GOTO CONCT
    IF &DATL = &SER THEN SET &COZ = &STR('&SER')
    IF &MONC = &STR(KATHER) | &CONC = &STR(KT) THEN SET +
        &COZ = &COZ&STR( 'INR105.KATHER.LOAD' )
    IF &CONC = &STR(TRACEGS) | &CONC = &STR(TR) THEN SET +
        &COZ = &COZ&STR( 'TSOSYS.TRACEGS7' )
    CONCT: WRITE FOLGENDE DATEIEN WERDEN ZUSAMMENGEFASST :
    WRITE '&DATL' &COZ
    WRITE &STR( ) &COS
    CONCAT ( '&DATL' &COZ &COS )
    COPY &DATF &PROG..FORT
    FORT77 &PROG. NOGO OBJECT(&PROG..OBJ) PRINT(*) LONGLVL(77) &DEB
    LINK &PROG..OBJ LET PRINT(*) LOAD('&DATL.(&GLIED)') &HEL
    WRITE MODUL '&DATL.(&GLIED.)' GESCHAFFEN
    COMPRESS '&DATL' NOLIST
    REDUCE '&DATL'
    ERASE &PROG..FORT
    ERASE &PROG..OBJ
    IF &ICA=&STR() THEN GOTO FIN
    CONTROL PROMPT MAIN NOMSG END(QUIT)
    SET &DATC=&SYSUID..&PROG..CLIST
    IF &SYSDSN('&DATC')=&OK THEN +
        EX '&SYSUID..CLIST(ADA)' '&DATC LIN(80)'
    EDIT '&DATC' OLD CLIST NUM
    RENUM 10 10
    DELETE 10 50
    IF &F1 = &STR() THEN +
    15 ALLOC DA('&DAT1') F(&F1) SHR REU
    IF &F2 = &STR() THEN +
    16 ALLOC DA('&DAT2') F(&F2) SHR REU

```

```

20 CALL '&DATL.(&GLIED.)'
IF &F1 = &STR() THEN 25 FREE DA('&DAT1')
IF &F2 = &STR() THEN 26 FREE DA('&DAT2')
LIST
END SAVE
WRITE AUFRUF IN '&DATC' GESPEICHERT
FIN: EXIT

```

RACLI

RACLI erzeugt einen KATALOG der eigenen RACF-Eintragungen und deponiert ihn - zwecks weiteren Handhabung - in die Datei '&SYSUID..MEMO'. RACLI Benötigt die Hilfsprozedur inr000.CLIST(ADA).

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG
WRITE
WRITE DIE RACF-ZUSTAENDE DES STAMMES &SYSUID. +
WERDEN IN DIE DATEI '&SYSUID..MEMO' GELISTET.
ERASE '&SYSUID..MEMO'
EX (ADA) '&SYSUID..MEMO LIN(80)'
ALLOC FILE(INTEX) DA('&SYSUID..MEMO') SHR REU
OPENFILE INTEX OUTPUT
SET &SYSOUTTRAP = 900
LISTDSD ID(&SYSUID) HISTORY ALL
SET I = 1
ERROR GOTO I2
DO WHILE &I <= &SYSOUTLINE
    SET LIN = &STR(&&SYSOUTLINE&I)
    SET INTEX = &STR(&LIN)
    PUTFILE INTEX
    I2: SET &I = &I+1
END
CLOSFILE INTEX
FREE FILE(INTEX)
EXIT

```

92/02/28

SERIE

SERIE erzeugt - mit Hilfe eines Glieder-Katalogs (s. Prozedur KATALOG) - aus einer gegliederten Datei eine serielle Datei.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)

Benötigte Hilfsdatei :

- inr000.MEMORY(SERIE)

Die Prozedurliste:

```

PROC 0 H0(-)
CONTROL PROMPT MAIN MSG END(STOP)
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE PROZEDUR      SERIE
WRITE AUS EINER GEGLIEDERTEN DATEI WIRD
WRITE - ANHAND EINES GLIEDER-KATALOGS &H0
WRITE EINE UNGEGLIEDERTE DATEI HERGESTELLT.
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(SERIE)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &N3 &M1 &NZ1 &NZ2 &NZ3
SET &KAMEL=&STR()
WRITE
WRITE ANGABEN ZUM KATALOG :
EX '&SYSUID..CLIST(NEVE)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &QUELLE=&TON
IF &M1=&STR(#) THEN DO
    EX '&SYSUID..CLIST(TAGOK)'
    SET &QUELLE=&TON(&M1)
    STOP
SET &KAMEL = &KAMEL &M1
WRITE SOLLEN DIE NAMEN DER GLIEDER MIT EINGEFUEGT WERDEN ? +
( "ENTER" = JA ) =:
READ &ANS
IF &ANS = &STR() THEN SET &GN=1
    ELSE SET &GN=0
IZ: WRITE ANGABEN ZUR ZIELDATEI :
SET &N1=&NZ1
SET &N2=&NZ2
SET &N3=&NZ3
EX '&SYSUID..CLIST(NEVE)' 'CQ(1)'
IF &N1 = &STR(#) THEN GOTO FIN
IF &SYSDSN('&TON') = OK THEN DO
    WRITE DIE DATEI &TON EXISTIERT SCHON!
    WRITENR. WOLLEN SIE ES UEBERSCHREIBEN ? =: ( "ENTER" = JA )
    READ &ANS
    IF &ANS = &STR() THEN GOTO IZ
    STOP
SET &ZIEL=&TON
EX '&SYSUID..CLIST(BEKI)' 'MEM(SERIE)'
WRITE
WRITE HERSTELLUNG DER SERIELLEN DATEI:
WRITE
ALLOC FILE(INTEX) DA('&QUELLE') SHR REU
OPENFILE INTEX INPUT
ERASE '&ZIEL'
ERROR GOTO I5
SET M = 0
DO WHILE &M <= 200
    GETFILE INTEX
    SET &SYSDVAL= &INTEX
    SET &M = &M+1
    IF &GN = 1 THEN MERGE '&QUELLE' &M '&ZIEL' NONUM NONUM2
    READDVAL &MEMB
    MERGE '&MEMB' 1 9999 '&ZIEL' NONUM NONUM2
    WRITE &M.. &MEMB &LASTCC

```

91/02/20

```

STOP
I5: CLOSFILE INTEX
ERROR OFF
WRITE
WRITE
WRITE &M GLIEDER IN '&ZIEL' ZUSAMMENGELEGT
WRITE
FREE FILE(INTEX)
WRITE KATALOG LOESCHEN ? ( "ENTER" = JA | # = &ZIEL ZUFUEGEN ) =:
READ &ANS
IF &ANS=&STR() THEN ERASE '&QUELLE'
IF &ANS=&STR(#) THEN MERGE '&QUELLE' 1 9999 '&ZIEL' NONUM NONUM2
REDUCE '&ZIEL'
FIN: EXIT

```

SPY

SPY erleichtert den Einstieg in eine SPEAKEASY(ZETA)-Sitzung (/4/). Eine Speicher-Datei (SPEAKEZ.DATA) und zwei Hilfsdateien werden bereitgestellt, die grafische Darstellung vorbereitet.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)

Benötigte Hilfsdatei :

- inr000.MEMORY(SPY)

Die Prozedurliste:

```

PROC 0 NAME(NAPLO) GLIED(LO)
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE
WRITE PROZEDUR      SPY - IV . ZETA          91/11/19
WRITE VOR- UND NACHBEREITUNG EINER SPEAKEASY SITZUNG
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(SPY)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &PRO1 &NAM1 &PRO2 &NAM2
SET &KAMEL=&STR()
WRITE SPEAKEASY-DATEI "MYKEPT" :
SET &N3=&STR(DATA)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
IF &N1 = &STR(#) THEN EXIT
SET &BUCH=&TON
SET &JOUR=&STR(&SYSUID..JOURNAL.DATA)
WRITE WEITERE HILFSDATEIEN :
SET &I=1
DO WHILE &I <= 2
    SET P=&&PRO&I
    SET Q=&&NAM&I
    WRITE DIE &I.. HILFSDATEI :
    SET &N1=&P
    SET &N2=&Q.

```



```

SET &N3=&STR(DATA)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
IF &N1 = &STR(#) THEN GOTO BEG
SET &DAT&I=&TON
SET &I=&I+1
END
BEG: IF &I = 1 THEN SET &KAMEL=&KAMEL &PRO2 &NAM2
EX '&SYSUID..CLIST(BEKI)' 'MEM(SPY)'
WRITE BEGINN DER PROZEDUR :
ALLOC FI(ADMGDF) DA('&SYSUID..ADMGDF') SHR REU
IF &BUCH=&STR() THEN +
  ALLOC FI(MYKEPT) DA('&BUCH') SHR REU
IF &DAT1=&STR() THEN +
  ALLOC F(FT15F001) DA('&DAT1') SHR REU
IF &DAT2=&STR() THEN +
  ALLOC F(FT16F001) DA('&DAT2') SHR REU
IF &SYSDSN('&JOUR')=OK THEN +
  EX '&SYSUID..CLIST(ADA)' '&JOUR LIN(133)'
ALLOC FI(JOURNAL) DA('&JOUR') SHR REU
ALLOC FI(ADMDEFS) DA('TSOSYS.ADMDEFS') SHR REU
ALLOC DA('IRE443.SPEAK90.PROCLIB') FI(PROCLIB) SHR
CALL 'IRE443.SPEAK90.ZETA.LOAD(SPEAK4)'
/* SPEASY
IF &SYSDSN('&JOUR')=OK THEN DO
  WRITE NACHBESSERUNG DER JOURNAL ? ( "ENTER"=JA ) =:
  READ &ANS
  IF &ANS=&STR() THEN GOTO FIN
  COPY '&JOUR' '&SYSUID..&NAME..DATA(&GLIED.)' NONUM
  ERASE '&JOUR'
  END
FIN: FREE FI(ADMDEFS ADMGDF MYKEPT)
IF &DAT1=&STR() THEN FREE DA('&DAT1')
IF &DAT2=&STR() THEN FREE DA('&DAT2')
EXIT

```

SS

SS Handhabt die Ausdruck-Datei 'inr000.SPF1.LIST', oder die Log-Datei 'inr000.SPFLOG3.LIST'. Die Möglichkeiten sind

- auflisten,
- kopieren,
- ausdrucken,
- löschen.

Seit es die Dateinamen 'tso000.inr000.spflog3.list' u. ä. nicht mehr gibt eigentlich eine überflüssige Prozedur.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(DRUQQ)
- inr000.CLIST(FEJ)
- inr000.CLIST(SUSA)

Benötigte Hilfsdatei :

- inr000.MEMORY(FEJ)

Die Prozedurliste:

```
PROC 0 QD(*)
CONTROL PROMPT MAIN MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE
WRITE HANDHABUNG DER SERVICE-DATEIEN SPF1 | SPFLOG3
IN: WRITE DATEI
WRITE      SPF1      : 1
WRITE      SPFLOG3   : 3
WRITE      KEINE     : #
READ &ANS
IF &ANS = &STR(#) THEN GOTO FIN
IF &ANS = 1 THEN SET &NADA=&STR(SPF1)
IF &ANS = 1 THEN SET &NADA=&STR(SPFLOG3)
SET &TON = &SYSUID..&NADA..LIST
IF &SYSDSN('&TON')= OK THEN DO
  WRITE NICHT VORHANDEN !
  GOTO IN
  QUIT
REDUCE '&TON'
MH: WRITE DIE DATEI '&TON'
WRITE      AUFLISTEN : L
WRITE      AUSDRUCKEN : P
WRITE      KOPIEREN  : C
WRITE      LOESCHEN  : #
WRITE      ENDE      : !
WRITE      NIX       : "ENTER"
READ &ANS
IF &ANS=&STR(!) THEN GOTO FIN
IF &ANS=&STR() THEN GOTO IN
IF &ANS=&STR(#) THEN DO
  ERASE '&TON'
  GOTO FIN
  QUIT
IF &ANS=&STR(L) THEN L '&TON'
IF &ANS=&STR(C) THEN DO
  COPY '&TON' CLIST.LIST NONUM +
  RECFM(FBA) LRECL(120) BLOCK(3120)
  WRITE '&TON' IST IN DER DATEI CLIST.LIST
  QUIT
IF &ANS=&STR(P) THEN DO
  EX (FEJ)
  EX (DRUQQ)
  EDIT &N3..CNTL OLD NUM
  260 PRINT TYPORG=PS,MAXFLDS=1&M1
  261 RECORD FIELD=(&N2)
  266 //
  EX (SUSA) '&N3.'
  QUIT
GOTO MH
FIN: EXIT
```

91/09/25

TAFEL

Einstieg in eine vereinfachte MAPPER-Prozedur. Die Erzeugung von "Metafiles" wird hier nicht unterstützt. Andererseits ist TAFEL ein tüchtiger Hausmann, temporäre Dateien werden bei Bedarf nicht nur angelegt, sondern nach Gebrauch auch beseitigt.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADU)
- inr000.CLIST(BEQI)
- inr000.CLIST(FEJ)
- inr000.CLIST(GLIEDER)
- inr000.CLIST(NEVE)
- inr000.CLIST(SUSA)
- inr000.CLIST(TAGOK)
- inr000.CLIST(TUKOR)

Neben der Hauptdatei - inr000.MAPPER.DATA - die die Tafeln enthält braucht man noch die folgenden Hilfsdateien :

- inr000.MEMORY(MAPPER)
- inr000.MEMORY(VERSAT)
- inr000.MEMORY(FEJ)

Die Prozedurliste:

```
PROC 0 V(T) DAT1(&SYSUID..MEMORY),+
      DAT2(&SYSUID..MAPPER.DATA),+
      DAT3(&SYSUID..MAPPER.TEMP),+
      DAT4(&SYSUID..MAPPER.ERRORS),+
      DAT6(&SYSUID..KFK),+
      DAT8(IRE443.CONFIG.ADMSYMBL),+
      DAT9(IRE443.MAPPER.LOAD)
/* *****
/* *   STAND: 25.01.1991   15.00 UHR   *
/* *   W. ZIMMERER      ,   TEL. 3764   *
/* *   BETRAECHTLICH VEREINFACHT , SEPT. 89 *
/* *   K. THURNAY      ,   TEL. 2477   *
/* *****
CONTROL PROMPT MAIN MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 M1 TON G
SET &OUTS=&STR(&SYSUID &SYSDATE &SYSTIME)
WRITE
WRITE MAPPER:   ERSTELLUNG EINER TEXT-TAFEL IM DIALOG   91/02/21
WRITE SITZUNG:  &OUTS ,   LOAD-MODUL: &DAT9.
WRITE
IF &SYSDSN('&DAT2')=&OK THEN DO
    WRITE DATEI '&DAT2'   FEHLT !
    EXIT
    QUIT
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(MAPPER)'
II: SET &SYSDVAL=&KAMEL
READDVAL &G &LG &BR &SX &SY &MEMB
IF &G = &STR() | &SX = &STR() | &SY = &STR() +
| &SX = &STR(0) | &SY = &STR(0) THEN DO
    SET &KAMEL=&STR(5 35.56 27.94 1.00 1.00 DUMMY)
    GOTO II
    QUIT
EX '&SYSUID..CLIST(TUKOR)'
```

```

SET &G=&N3
SET &KAMEL=&STR(&G 0 0 &LG &BR &SX &SY)
SET &TON=&DAT2
SET &M1=&MEMB
ALLOC FI(ADMDEFS) DA('TSOSYS.ADMDEFS') SHR REU
BI: WRITE NAME DER BILDТАFEL :
EX '&SYSUID..CLIST(TAGOK)'
SET &MEMB=&M1
IF &M1=&STR(!) THEN GOTO FIN
EX '&SYSUID..CLIST(BEKI)' 'MEM(MAPPER)'
IF &SYSDSN('&DAT4')=&OK THEN +
  EX '&SYSUID..CLIST(ADA)' '&DAT4 LIN(133)'
IF &SYSDSN('&DAT3')=&OK THEN +
  EX '&SYSUID..CLIST(ADA)' '&DAT3 LIN(80)'
ALLOC FI(FT09F001) DA('&DAT2.(&M1)') REU SHR
ALLOC FI(FT10F001) DA('&DAT3.') REU SHR
CALL '&DAT9.(CONVER)''
ALLOC FI(FT13F001) DA('&DAT4.') REU SHR
ALLOC FI(FT21F001) DA('&DAT3.') REU SHR
ALLOC FI(FT22F001) DA('&DAT6.') REU SHR
ALLOC FI(FT05F001) DA('&DAT1.(MAPPER)') REU SHR
IF &G = &STR(3) THEN CALL '&DAT9.(MAPPERGA)''
ELSE DO
  ALLOC FI(ADMGDF) DA('&SYSUID..ADMGDF') SHR REU
  ALLOC FI(ADMSYMBL) DA('&DAT8.') REU SHR
  CALL '&DAT9.(MAPPERNW)''
  FREE FI(ADMSYMBL)
  QUIT
FREE FI(FT05F001,FT09F001,FT10F001,FT13F001,FT21F001,FT22F001)
IV: WRITE VORGEHEN / &V / =: ! = ENDE
WRITE          T = NEUER TAFEL
WRITE          F = FEHLERNACHRICHTEN
WRITE          V = VERSATEC - BILD
WRITE          C = VERSA-COLOR BILD
WRITE          L = BILD AM 3820 DRUCKER
READ &ANS
IF &ANS &= &STR() THEN SET &V = &ANS
IF &V = &STR(!) THEN GOTO FIN
IF &V = &STR(T) THEN GOTO BI
IF &V = &STR(C) | &V = &STR(L) | &V = &STR(V) THEN GOTO HC
IF &V = &STR(F) THEN LIST '&DAT4.' 6 999 COL(1:72) NONUM
GOTO IV
HC: WRITE VORBEREITUNG ZUR BILDAUSGABE :
IF &V = &STR(L) THEN GOTO ED
SET EA=1
SET &KAMEL=&STR( )
EX '&SYSUID..CLIST(BEKI)' 'MEM(VERSAT)''
WRITE BITTE DIE FOLGENDE ANGABEN MIT DEZIMALPUNKT !
WRITE
WRITE BILDLAENGE IN DER X-RICHTUNG IN CM / &LG / =:
READ &ANS
IF &ANS &= &STR() THEN SET &LG = &STR(&ANS&STR(000))
WRITE BILDBREITE IN DER Y-RICHTUNG IN CM / &BR / =: +
( MAXIMUM = 55 CM )
READ &ANS
IF &ANS &= &STR() THEN SET &BR = &STR(&ANS&STR(000))
SET &SL=&SUBSTR(1:5,&LG)
SET &SB=&SUBSTR(1:5,&BR)
ED: EX '&SYSUID..CLIST(FEJ)' 'R(2000)''

```

```

EDIT '&SYSUID..&N3..CNTL' OLD CNTL NUM
IF &V = &STR(V) THEN +
120 //SWWSLOT EXEC FVG,NAME=&STR(MAPVERNW),PLOU=V
IF &V = &STR(C) THEN +
120 //COLOPLOT EXEC FVG,NAME=&STR(MAPVSCOL)
IF &V = &STR(L) THEN +
120 //IBM3820 EXEC FVG,NAME=&STR(MAPP3820)
130 //G.STEPLIB DD
132 // DD DISP=SHR,DSN=&DAT9.
IF &V = &STR(L) THEN 133 //*G.VRFOUT DD DUMMY
150 //G.FT13F001 DD UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(CYL,(2,2))
160 //G.FT21F001 DD DISP=SHR,DSN=&DAT3.
170 //G.FT22F001 DD DISP=SHR,DSN=&DAT6.
SET &SYSSCAN=1
IF &V = &STR(L) THEN 200 //G.TEST0001 DD SYSOUT=D
IF &V = &STR(L) THEN DO
  200 //G.PLOTPARM DD *
  210 &&PLOT +
  XFACT=1.0,YFACT=1.0,PXMAX=50.,VXMAX=50.,WXMAX=50. &&END
  QUIT
220 //G.SYSIN DD *
IF &V = &STR(L) THEN 230 1 0 0 &SL &SB &SX &SY
IF &V = &STR(V) THEN 230 6 0 0 &SL &SB &SX &SY
IF &V = &STR(C) THEN 230 8 0 0 &SL &SB &SX &SY
SET &SYSSCAN=16
300 //*
310 //PRINT EXEC PGM=IEBPTPCH,COND=EVEN
320 //SYSPRINT DD DUMMY
330 //SYSUT1 DD DISP=SHR,DSN=&DAT3.
340 //SYSUT2 DD SYSOUT=W
350 //SYSIN DD *
360 PRINT TYPORG=PS,MAXFLDS=1
370 RECORD FIELD=(80,1,,10)
999 //
EX '&SYSUID..CLIST(SUSA)' '&N3'
REDUCE '&DAT3.'
GOTO FIH
FIH: ERASE '&DAT3.'
FIH: SET &KAMEL=&G &LG &BR &SX &SY &MEMB
SET EA=1
EX '&SYSUID..CLIST(BEKI)' 'MEM(MAPPER)'
ALLOC FI(FT05F001) DA(*)
FREE FI(ADMDEFS)
ERASE '&DAT4.'
EXIT

```

TSOKO

TSOKO lenkt die - von einem TSO-Kommando erzeugte - Ausgabezeilen um vom Bildschirm in die Datei 'inr000.MEMO'. TSOKO ist im Augenblick auf eine Ausgabe, die aus höchstens 3000 Zeilen besteht eingerichtet. TSOKO kann man mit Hilfe der **Attn**-Taste jederzeit beenden.

Eingabegrößen der Prozedur: Mit der Schlüsselwortparameter **Q** kann man die Art der Ausgabe steuern :

- **Q(HELP)** erzeugt die Beschreibung des Kommandos

- Q(sonst) (Voreinstellung) erzeugt die eigentliche Ausgabe
- COMM enthält den Namen des TSO-Kommandos,
- P1 erlaubt es, Eingabegrößen des Kommandos als Zeichenkette zwischen Hochkommatas weiterzureichen, z.B. :

```
TSOKO LISTALC P1('ST H')
```

Obiger Befehl erzeugt z. B. folgende MEMO-Datei :

```

*****
***                ***
***      LISTALC   ***
***                ***
*****

--DSORG--CREATED---EXPIRES---SECURITY--DDNAME---DISP--
TSOSYS.DCQOLIB
  PO    10/31/90  00/00/00  NONE      STEPLIB  KEEP
SYS7.VSCOMP
  PO    07/29/91  12/26/99  NONE                KEEP
SYS7.VSLOAD
  PO    07/29/91  12/26/99  NONE                KEEP
SYS1.HELP
  PO    09/06/91  12/31/99  NONE      SYSHELP  KEEP
TSOSYS.ECF110.MVSSERV.PARM
  PS    04/22/88  00/00/00  NONE      CHSPARM  KEEP
TSOSYS.ECF110.MVSSERV.DUMP
  PS    04/22/88  00/00/00  NONE      CHSABEND KEEP
SYS7.VSFORT
  PO    07/29/91  12/26/99  NONE      SYSLIB   KEEP
SYS2.ACRITH
  PO    01/30/90  12/31/99  NONE                KEEP
SYS2.HARWELL
  PO    01/30/90  00/00/00  NONE                KEEP
SYS2.FORTLIB
  PO    10/31/90  00/00/00  NONE                KEEP
.....
.....
  PO    05/16/91  00/00/00  NONE      SYS00031 KEEP
INR105.CLIST.TEXT
  PO    02/29/92  00/00/00  NONE      ISP08600 KEEP,KEEP
INR105.MEMO
  PS    03/05/92  00/00/00  NONE      SYS00035 CATLG
INR105.MEMO
  PS    03/05/92  00/00/00  NONE      INTEX    KEEP

```

Vorsicht! Beim Help-Ausgabe soll man P1 nicht mitgeben! Z. B.

```
TSOKO LISTALC Q(HELP) P1('ST H')
```

führt zum Stillstand!

Mit TSOKO kann man versuchen, auch selbstgebaute CLIST-s zu melken, aber die Ergebnisse sind nicht immer vorhersehbar und befriedigend! Lassen Sie bei solchen Versuchen die **Attn**-Taste nicht aus der Auge!

TSOKO Benötigt nur die Hilfsprozedur `inr000.CLIST(ADA)`.

Die Prozedurliste:

```

PROC 1 COMM Q() P1() IH(0) +
ZA('***      ') ZE('      ***') ZV('*****') ZL('      ')
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE PROCEDUR TSOKO
ATTN DO
    WRITE ENDE DURCH ATTENTION
    EXIT
    END
SET T=&STR()
IF &Q = &STR(HELP) THEN SET TX=&STR(BESCHREIBUNG)
ELSE DO
    SET TX=&STR(AUSGABE)
    SET Q=&STR()
    SET IH=1
    END
WRITE &Q &COMM &P1
SET L=&LENGTH(&COMM)
SET V=&SUBSTR(1:&L,&ZV)
SET L=&SUBSTR(1:&L,&ZL)
ERASE '&SYSUID..MEMO'
EX (ADA) '&SYSUID..MEMO LIN(80)'
ALLOC FILE(INTEX) DA('&SYSUID..MEMO') SHR REU
CONTROL PROMPT MAIN MSG ASIS
OPENFILE INTEX OUTPUT
SET &SYSOUTTRAP = 3000
WRITE DIE &TX DES KOMMANDOS &COMM +
KOMMT IN DIE DATEI '&SYSUID..MEMO'
IF &IH = 1 THEN DO
    WRITE
    WRITE DIE PROZEDUR KANN MAN DURCH "ATTENTION" BENDEN
    WRITE
    IF &SISISPF = &STR(ACTIVE) THEN DO
        SET &T = &STR(TSO)
    END
END
&T &Q &COMM &P1
SET INTEX = &STR()
PUTFILE INTEX
SET INTEX = &STR(&ZL&ZV&V&ZV)
PUTFILE INTEX
SET INTEX = &STR(&ZL&ZA&L&ZE)
PUTFILE INTEX
SET INTEX = &STR(&ZL&ZA&COMM&ZE)
PUTFILE INTEX
SET INTEX = &STR(&ZL&ZA&L&ZE)
PUTFILE INTEX
SET INTEX = &STR(&ZL&ZV&V&ZV)
PUTFILE INTEX
SET INTEX = &STR()
PUTFILE INTEX
SET I = 1

```

91/02/27

```

ERROR GOTO I2
DO WHILE &I <= &SYSOUTLINE
  SET LIN = &STR(&&SYSOUTLINE&I)
  SET INTEX = &STR(&LIN)
  PUTFILE INTEX
  I2: SET &I = &I+1
  END
CLOSFIL INTEX
EXIT

```

WEG

WEG unterstützt den Benutzer beim Löschen einer Datei mit der Prozedur **ERASE** sowohl im Vordergrund (Institus-Platten), als auch im Massenspeicher. WEG bietet den Vorteil, daß sie eventuelle Tippfehler im Datei-Namen erkennt und bei der Korrektur hilft.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)

Benötigte Hilfsdatei :

- inr000.MEMORY(CORE)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE DATEIEN AUF DER VORDERGRUND-PLATTEN SOWIE
WRITE IM MASSENSPEICHER WERDEN GELOESCHT.
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(CORE)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &N3
SET &KAMEL=&STR()
SET &V=&STR()
BE: EX '&SYSUID..CLIST(NEVE)' 'CQ(1)'
IF &TON = &STR() THEN GOTO FIN
SET &TON = &STR('&TON.')
IF &SUBSTR(1:1,&N1) = &STR(I) THEN SET &D = &STR(B)
IF &SUBSTR(1:1,&N1) = &STR(T) THEN SET &D = &STR(T)
SET &TX = &D&STR(L)
VN: WRITE NUMMER DER VERSION / &V / =:
WRITE          ( ENTER = DATEI IM VORDERGRUND
WRITE          ? = LISTE DER VERSIONEN
WRITE          ! = NEUE DATEI
WRITENR       # = ENDE )
READ &ANS
IF &ANS = &STR(#) THEN GOTO FIN
IF &ANS = &STR(!) THEN DO
  SET &TON = &STR()
  GOTO BE
END
IF &ANS = &STR(?) THEN DO
  &TX &TON

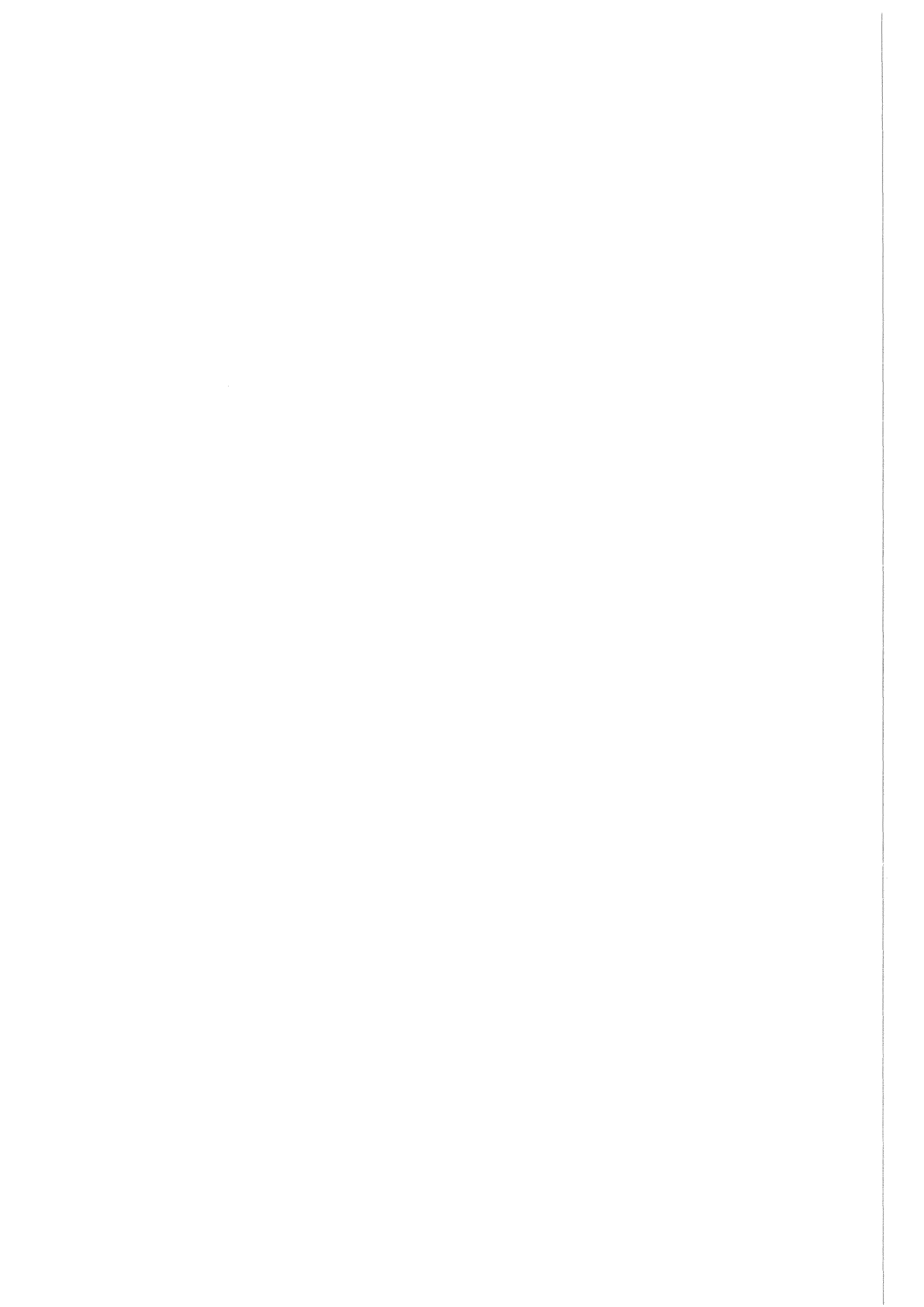
```

91/02/22


```

GOTO VN
END
SET &V = &ANS
IF &V = &STR() THEN SET &TOD = &STR( &TON )
IF &V ^= &STR() THEN SET &TOD = &STR( &TON VER(&V) )
WRITE
WRITE SOLL DIE DATEI &TOD
WRITE
WRITENR      GELOESCHT WERDEN ? =: ( # = JA | SONST NEIN )
READ &ANS
IF &ANS=&STR(#) THEN DO
    ERASE &TOD
    IF &V = &STR() && &LASTCC = 0 THEN +
    WRITE DATASET &TOD ERASED
    WRITE
    END
GOTO VN
FIN: EX '&SYSUID..CLIST(BEKI)' 'MEM(CORE)'
EXIT

```



G.S.-Prozeduren

Der Umgang mit diesen Prozeduren setzt Grundkenntnisse über das Graphische System (G.S.) voraus. Für eine Beschreibung des Systems sollte man sich das GS-Handbuch /5/ bei Frau Friedle, HDI besorgen. Die im Folgenden beschriebenen Prozeduren setzen außerdem voraus, daß der Benutzer zumindest ein GS-Buch besitzt und daß die Datei 'inr000.ADMGDF' bereit steht.

Anmerkung: Alle folgenden Prozeduren arbeiten - verdeckt - in der G.S.-Umgebung. Bei einer Fehleingabe (z.B. eine aus Versehen eingegebene " / ") erscheint somit das dortige Eingabe-Zeichen **GS**. Durch eintippen von **ENDGS** kann man sich wieder befreien.

BRAUS

BRAUS hilft - im Rahmen eines Dialogs - dem Benutzer, Bilder eines GS-Buches auszuwählen und sie durch einen der Stapel-Geräte zeichnen zu lassen.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(ALOGS)
- inr000.CLIST(BEKI)
- inr000.CLIST(FEJ)
- inr000.CLIST(GLIEDER)
- inr000.CLIST(NEVE)
- inr000.CLIST(SUSA)
- inr000.CLIST(TAGOK)
- inr000.CLIST(TUKOR)

Benötigte Hilfsdateien :

- inr000.MEMORY(BRAUS)
- inr000.MEMORY(BOOKS)

Die Prozedurliste:

```
PROC 0 NB(0) IZ1(0) IZ2(0) IZ3(0) IZ4(0)
CONTROL PROMPT MAIN MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 BIN TON AP
WRITE
WRITE GS-PROZEDUR BRAUS
WRITE BILDER, DIE IN EINEM GS-BUCH GESPEICHERT SIND
WRITE WERDEN AN EINEM STAPEL-PLOTTER GEZEICHNET.
SET &EA=0
EX '&SYSUID..CLIST(ALOGS) '
SET &GOBLIN=&KAMEL
SET &SYSDVAL=&KAMEL
READDVAL &SIZE &GSB &UFB &SF &ZUB
IF &SIZE=&STR(0) THEN EXIT
SET &EA=0
EX '&SYSUID..CLIST(BEKI) ' 'MEM(BRAUS) '
SET &L = &LENGTH(&GSB)
SET &PRJ = &SUBSTR(1:6,&GSB)
SET &BCH = &SUBSTR(8:&L,&GSB)
SET &SYSDVAL=&KAMEL
READDVAL &GER &BIN &QU &DIN &FORM
```

91/02/20

```

WRITE ZEICHENGERAET / &GER / =:
WRITE "VER" : VERSATEC
WRITE "XYN" : XYNETICS
WRITE "VCP" : VERSACOLOR FARBIG
WRITE "VMP" : VERSACOLOR SCHWARZ-WEISS
READ &ANS
IF &ANS ^= &STR() THEN SET &GER = &SUBSTR(1:3,&ANS)
IF &GER = &STR(VER) THEN DO
    SET &KG=1
    SET &QU=&STR(V)
    SET &ESP=&STR(GS7,PLOU=&QU)
    QUIT
IF &GER = &STR(XYN) THEN DO
    SET &KG=2
    WRITE AUSFUEHRUNG / &QU / =: ( "ENTER"=JA | X=KULI | Y=TUSCHE )
    READ &ANS
    IF &ANS ^= &STR() THEN SET &QU = &ANS
    SET &ESP=&STR(GS7)
    QUIT
IF &GER = &STR(VCP) | &GER = &STR(VMP) THEN DO
    IF &GER = &STR(VCP) THEN SET &KG=3
    IF &GER = &STR(VMP) THEN SET &KG=4
    SET &QU=&STR(X)
    SET &ESP=&STR(GS7,PLOU=&QU)
    QUIT
SET &IR = 2
WRITE BILDNAME IN DIE ZEICHNUNG EINTRAGEN ? ( "ENTER"=JA )
READ &ANS
IF &ANS ^= &STR() THEN SET &IR = 1
WRITE BEGINN DER PROZEDUR
SET &TN1=&STR()
SET &TN2=&STR()
SET &TN3=&STR()
SET &TN4=&STR()
SET &TN5=&STR()
SET &TDI=&STR()
SET &TF0=&STR()
SET &TPX=&STR()
SET &TPY=&STR()
BI: WRITE NAME DES BILDES :
SET &TON=&GSB
SET &LAB=&BIN
EX '&SYSUID..CLIST(TAGOK)'
IF &BIN=&STR(!) THEN DO
    SET &BIN=&LAB
    GOTO FIA
    QUIT
BF: WRITENR BILDFENSTER DIN-A / &DIN / =:
    READ &ANS
    IF &ANS ^= &STR() THEN DO
        SET &DIN=&ANS
        IF &DATATYPE(&DIN) ^= NUM | &DIN < 1 | &DIN > 6 THEN GOTO BF
    QUIT
ST: WRITENR STELLUNG / &FORM / =: ( H=HOCH | Q=QUER )
READ &ANS
IF &ANS ^= &STR() THEN SET FORM=&SUBSTR(1,&ANS)
IF &FORM ^= H && &FORM ^= Q THEN GOTO ST
IF &FORM = H THEN SET QUER=0
    ELSE SET QUER=1

```

```

WRITE BILDKONTROLLE AM SCHIRM
INITGS -&SIZE
OW &AP
RPIB &BIN B1
SET &EFE=&STR(A4)&FORM
DWI 0 0 &EFE
ROUT
ENDGS
WRITENR AUSGEBEN ? ( "ENTER"=JA | ALLES ANDERE=NEIN ) =:
READ &ANS
IF &ANS=-&STR() THEN GOTO BI
SET NB=&NB+1
  SET &IS1=&IZ1
  SET &IS2=&IZ2
  SET &IS3=&IZ3
  SET &IS4=&IZ4
  SET &I = &IZ4
  IF &I > &IZ3 THEN SET &I=&IZ3
  IF &I > &IZ2 THEN SET &I=&IZ2
  IF &I > &IZ1 THEN SET &I=&IZ1
  SET &I = &I+1
L1: IF &DIN=6 | &DIN=5 THEN SET &MU=0
  IF &DIN=4 | &DIN=3 THEN SET &MU=1
  IF &DIN<=2 THEN SET &MU=3
  SET &IZM=50
  IF &GER=&STR(XYN) THEN SET &IZM=7
L2: SET &NU=&I+&MU
  IF &GER=&STR(XYN) && &NU > &IZM THEN GOTO VL
  IF &IZ1>=&I THEN GOTO L5
  IF &DIN=6 THEN GOTO L4
  IF &IZ2>=&I THEN GOTO L5
  IF &DIN>3 THEN GOTO L3
  IF &IZ3>=&I | &IZ4>=&I THEN GOTO L10
  SET &IZ4=&NU
  SET &IZ3=&NU
L3: SET &IZ2=&NU
L4: SET &J=1
  SET &IZ1=&NU
  GOTO S1
L5: IF &DIN<4 THEN GOTO L10
  IF &IZ2>=&I THEN GOTO L7
  IF &DIN=6 THEN GOTO L6
  IF &IZ3>=&I THEN GOTO L7
  SET &IZ3=&NU
L6: SET &J=2
  SET &IZ2=&NU
  GOTO S1
L7: IF &IZ3>=&I THEN GOTO L9
  IF &DIN=6 THEN GOTO L8
  IF &IZ4>=&I THEN GOTO L9
  SET &IZ4=&NU
L8: SET &J=3
  SET &IZ3=&NU
  GOTO S1
L9: IF &DIN<6 | &IZ4>=&I THEN GOTO L10
  SET &J=4
  SET &IZ4=&NU
  GOTO S1
L10: SET &I=&I+1

```

```

GOTO L2
VL: WRITE NICHT GENUG PLATZ !
    SET &NB=&NB-1
    SET &IZ1=&IS1
    SET &IZ2=&IS2
    SET &IZ3=&IS3
    SET &IZ4=&IS4
    SET &I=-1
S1: IF &GER/= &STR(XYN) THEN GOTO S3
    IF &I < 0 THEN GOTO BI
    WRITE Z1/Z1M = &IZ1/&IZM , Z2/Z2M = &IZ2/&IZM ,
    WRITE Z3/Z3M = &IZ3/&IZM , Z4/Z4M = &IZ4/&IZM .
S3: WRITE &NB.. BILD = &BIN VERLEGT
    IF &NB > 5 THEN GOTO S4
    SET &TN1=&TN1&STR('&BIN&STR(',')
    GOTO S9
S4: IF &NB > 10 THEN GOTO S5
    SET &TN2=&TN2&STR('&BIN&STR(',')
    GOTO S9
S5: IF &NB > 15 THEN GOTO S6
    SET &TN3=&TN3&STR('&BIN&STR(',')
    GOTO S9
S6: IF &NB > 20 THEN GOTO S7
    SET &TN4=&TN4&STR('&BIN&STR(',')
    GOTO S9
S7: SET &TN5=&TN5&STR('&BIN&STR(',')
S9: SET &TFO=&TFO&QUER&STR(',')
    SET &TDI=&TDI&DIN&STR(',')
    SET &TPX=&TPX&I&STR(',')
    SET &TPY=&TPY&J&STR(',')
    IF &DIN=1 THEN GOTO FIA
    IF &IZ1 >= &IZM && &IZ2 >= &IZM && +
    &IZ3 >= &IZM && &IZ4 >= &IZM THEN DO
        WRITE DAS BLATT IST VOLL !
        GOTO FIA
    QUIT
IF &NB<24 THEN GOTO BI
WRITE DIE HOECHSTMUEGLICHE BILDZAHL (24) IST ERREICHT !
FIA: SET &KAMEL = &GOBLIN
EX '&SYSUID..CLIST(ALOGS)'
SET &KAMEL = &GER &BIN &QU &DIN &FORM
EX '&SYSUID..CLIST(BEKI)' 'MEM(BRAUS)'
IF &NB=0 THEN GOTO FIN
WRITE AUFTRAGSBEARBEITUNG :
EX '&SYSUID..CLIST(FEJ)' 'R(4000)'
EDIT '&SYSUID..&N3..CNTL' OLD CLIST NUM
120 //PICTURE EXEC F7CLG,PARM.C='LANGLVL(77)',
130 // PARM.L='MAP,LIST',PLOT=&ESP
140 //C.SYSIN DD *
190 C    HAUPTPROGRAMM ZUM AUSGABE VON BIS ZU 24 GESPEICHERTEN
200 C    GS-BILDERN AUF EINEM DER ZEICHENGERAETE IN
210 C    DIN-A-FORMAT.    STAND JUNI ,1988 .
220    DIMENSION X(2),Y(2),U(2),V(2),ZHF(2),ZPF(2),FF(2),FT(2)
230    =,FY(2),FG(4),GG(4),DAL(7),INFO(8),BIN(24),NID(24)
240    =,KBF(24),KPX(24),KPY(24)
250    CHARACTER*8 INFO,BCH,BIN,FG,FF,FT, GG*4,PRJ*6,BIM*9
260    =,TXT*26,FTA*25,FY*16,FYE*45,FFA*22,FFE*38
270    COMMON /LGS/ IRET
280    DATA ZUF/.103585/, ZPF/2.5,-1./, ZHF/2.5E-2,9.5E-3/

```

```

290     =,ZOLL/2.54/, DAL/33.1062,23.4096,16.5531
300     =,11.7048,8.27654,5.8524,4.13827/
310     DATA FG/'VERSATEC', 'XYNETICS', 'VERS.COL', 'VERS.UNI'/
320     =,GG/'VER_', 'XYN_', 'VCP_', 'VMP_'/
330     =,FTA/'(7H FORMAT,13X,7H= DIN A, '/
340     =,FT/'I2,2H Q)', 'I2,2H H)'/
350     =,FY/'(8H X0 , Y0,12X,', '(8H DX , DY,12X, '/
360     =,FYE/'3H= (,F8.3,2H ,,F8.3,2H ),5X,11H( CM , CM ) )'/,FFA
370     =/'(18H0* * * FEHLER BEI ', FF/'5H RPIB', '5H OUT '/
380     =,FFE/'7H . BILD,A9,9H . IRET =,Z8,6H * * *)'/
390     DATA PRJ/'PRJ'/,BCH/'&BCH'/,KG/'&KG'/,NB/'&NB'/,
(3IN(I),I=1,&NB)
400     =/&SUBSTR(1:&LENGTH(&TN1),&TN1)
IF &TN2=&STR() THEN +
410     =,&SUBSTR(1:&LENGTH(&TN2),&TN2)
IF &TN3=&STR() THEN +
420     =,&SUBSTR(1:&LENGTH(&TN3),&TN3)
IF &TN4=&STR() THEN +
430     =,&SUBSTR(1:&LENGTH(&TN4),&TN4)
IF &TN5=&STR() THEN +
440     =,&SUBSTR(1:&LENGTH(&TN5),&TN5)
450     =/,IR/'&IR/
460     =,(NID(I),I=1,&NB)/&SUBSTR(1:&LENGTH(&TDI),&TDI)/
470     =,(KBF(I),I=1,&NB)/&SUBSTR(1:&LENGTH(&TFO),&TFO)/
480     =,(KPX(I),I=1,&NB)/&SUBSTR(1:&LENGTH(&TPX),&TPX)/
490     =,(KPY(I),I=1,&NB)/&SUBSTR(1:&LENGTH(&TPY),&TPY)/
500     CALL JOBINF(INFO)
510     INFO(3)(1:1) = ' '
520     INFO(4)(4:8) = ' '
530     INFO(6)(1:2) = ' '
540     WRITE(6,'(10H DATUM : ,3A8,2X,2A8,11H , '//FG(KG)
550     = //',9H-PLOTTER.)')
560     = INFO(3),INFO(4),INFO(6),INFO(7),INFO(8)
570     WRITE(6,'(23H BILDER AUS DEM BUCH +,A8,1H.,A8,1H+ )')
580     = PRJ,BCH
590     CALL INITFG(IRET,-&SIZE)
600     CALL OW(GG(KG))
610     DO 111 I=1,24
620     BIM=' '
630     DO 113 J=1,8
640     IF(BIN(I)(J:J) .EQ. ' ') GO TO 114
650     BIM(J:J)=BIN(I)(J:J)
660     113 CONTINUE
670     114 CALL RPIB(BIM,'BUCH_')
680     IF(IRET .EQ. 0) GO TO 120
690     WRITE(6,FFA//FF(1)//FFE) BIN(I),IRET
700     GO TO 110
710     120 X(1)=0.787402
720     Y(1)=0.787402
730     IF(NID(I) .GT. 1) GO TO 130
740     IF(KG .EQ. 2) THEN
750     DX=DAL(1)
760     DY=DAL(2)
770     IF(KBF(I) .EQ. 1) WINK=0.0
780     IF(KBF(I) .NE. 1) WINK=90.
790     GO TO 140
800     ELSE
810     NID(I)=2
820     GO TO 130

```

```

830     ENDIF
840 130 X(1)=(KPX(I)-1)*DAL(6)*(1.+ZUF)+X(1)
850     Y(1)=(KPY(I)-1)*DAL(7)*(1.+ZUF)+Y(1)
860     IF(MOD(NID(I),2) .NE. 0) THEN
870         DX=DAL(NID(I)+1)
880         DY=DAL(NID(I))
890         IF(KBF(I) .NE. 1) WINK=0.0
900         IF(KBF(I) .EQ. 1) WINK=90.
910     ELSE
920         DX=DAL(NID(I))
930         DY=DAL(NID(I)+1)
940         IF(KBF(I) .EQ. 1) WINK=0.0
950         IF(KBF(I) .NE. 1) WINK=90.
960     ENDIF
970 140 WRITE(6, '(9H0BILDNAME,11X,1H=,A9,I29)') BIN(I),I
980     IF(KBF(I) .EQ. 1) WRITE(6,FTA//FT(1)) NID(I)
990     IF(KBF(I) .NE. 1) WRITE(6,FTA//FT(2)) NID(I)
1000    WRITE(6,FY(1)//FYE) ZOLL*X(1),ZOLL*Y(1)
1010    WRITE(6,FY(2)//FYE) ZOLL*DX,ZOLL*DY
1020    DU=DX*ZUF
1030    DV=DY*ZUF
1040    X(2)=X(1)+DX
1050    Y(2)=Y(1)+DY
1060    IF(WINK .NE. 0.) THEN
1070        WRITE(6, '(35H DAS BILD WURDE UM 90 GRAD GEDREHT)')
1080        CALL ROT(0.5*DX,0.5*DX,WINK)
1090        AH=DX*0.025
1100        ZH=DX*ZHF(IR)
1110        XT=X(2)+ZPF(IR)*ZH
1120        YT=Y(2)-21.6*ZH
1130        CALL DWI(X(1),Y(1),X(2),Y(2))
1140    ELSE
1150        CALL SHI(0.0,0.0,0.0,DV)
1160        AH=DY*0.025
1170        ZH=DY*ZHF(IR)
1180        YT=Y(1)+DV-ZPF(IR)*ZH
1190        XT=X(2)-21.6*ZH
1200        CALL DWI(X(1),Y(1)+DV,X(2),Y(2)+DV)
1210    ENDIF
1220    CALL OUT
1230    IF(IRET .NE. 0) WRITE(6,FFA//FF(2)//FFE) BIN(I),IRET
1240    CALL DPI('*_')
1250    CALL PI('LABEL_')
1260    TXT=PRJ//'. '//BCH//'('//BIN(I)//')_'
1270    U(1)=X(1)+AH
1280    V(1)=Y(1)
1290    U(2)=X(2)+DU
1300    V(2)=Y(2)+DV
1310    X(2)=U(2)
1320    Y(2)=V(2)-AH
1330    CALL SCO
1340    CALL VEC(X,Y,U,V,2)
1350    CALL STRO(XT,YT,TXT,ZH,WINK)
1360    CALL ECO
1370    CALL DWI(X(1),Y(1),U(2),V(2))
1380    CALL OUT
1390    IF(IRET .NE. 0) WRITE(6,FFA//FF(2)//FFE) 'LABEL',IRET
1400 110 IF(NB .EQ. 1) GO TO 900
1410 111 CALL DPI('LABEL_')

```



```

1420 900 CALL ENDGS(MAXS)
1430 WRITE(6,'(27H0GROESSTER SPEICHERBEDARF =,I4
1440 = ,8H K BYTES)') MAXS
1450 STOP
1460 END
1490 //L.SYSPRINT DD SYSOUT=*
1500 /**STEPLIB DD DISP=SHR,DSN=SYS7.VSLOAD
IF &GER = &STR(XYN) THEN DO
SET &IZM=&IZ1
IF &IZ2 > &IZM THEN SET &IZM=&IZ2
IF &IZ3 > &IZM THEN SET &IZM=&IZ3
IF &IZ4 > &IZM THEN SET &IZM=&IZ4
SET &XMX = 7*&IZM
IF &XMX<50 THEN SET &XMX=50
QUIT
IF &GER = &STR(XYN) THEN DO
SET &IDENT=&QU&SUBSTR(4:6,&PRJ)&N2
1550 //G.PLOTTAPE DD UNIT=T0800,LABEL=(,NL),DCB=DEN=2,+
VOL=SER=&IDENT
QUIT
ELSE DO
1510 /**VRFOUT DD DUMMY
1520 //G.PLOTPARM DD *
1530 &&PLOT LUNIT=26, &&END
1540 //G.FT26F001 DD SYSOUT=*
QUIT
1600 //G.BUCH DD DSN=&GSB,DISP=SHR
IF &UFB = &STR(#) THEN +
1610 //G.UFOLIB DD DSN=&UFB,DISP=SHR
1620 //G.COMM DD SYSOUT=*
2000 //
LIST 390 490
LIST 1520 2000
EX '&SYSUID..CLIST(SUSA)' '&N3.'
FIN: EXIT

```

IMMANI

IMMANI verwaltet GS-Bücher: Bilder werden unter die Lupe genommen, umgefärbt, verdreht, in eine GDF-Datei geschickt, gelöscht oder umbenannt.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(ALOGS)
- inr000.CLIST(BEKI)
- inr000.CLIST(GLIEDER)
- inr000.CLIST(NEVE)
- inr000.CLIST(TAGOK)
- inr000.CLIST(TUKOR)

Benötigte Hilfsdateien :

- inr000.MEMORY(IMMANI)
- inr000.MEMORY(BOOKS)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 BIN TON AP
WRITE
WRITE GS-PROZEDUR IMMANI
WRITE VERWALTUNG EINES GS-BUCHES :
WRITE # BILDER UMBENENNEN ,
WRITE # BILDER VOM ZOLL*ZOLL AUF CM*CM UMWANDELN,
WRITE # BILDER IN EINE GDF-DATEI ABLEGEN,
WRITE # NICHT MEHR BENOETIGTE BILDER LOESCHEN.
SET &EA=0
EX '&SYSUID..CLIST(ALOGS)'
SET &SYSDVAL=&STR(&KAMEL)
SET &GOBLIN=&KAMEL
READDVAL &SIZE &GSB &UFB &SF &ZUB
IF &SIZE=&STR(0) THEN EXIT
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(IMMANI)'
SET &SYSDVAL=&STR(&KAMEL)
READDVAL &BIN &FORM &XLU &YLU &XRO &YRO
WRITE BEGINN DER PROZEDUR
BI: WRITE NAME DES BILDES :
SET &TON=&GSB
SET &LAB=&BIN
EX '&SYSUID..CLIST(TAGOK)'
IF &BIN=&STR(!) THEN DO
    SET &BIN=&LAB
    GOTO FIN
    QUIT
SET &V = &STR()
INITGS -&SIZE
OW &AP
RPIB &BIN B1
MP: LA
WRITE
WRITE VORGEHEN =: B = NEUES BILD
WRITE G = GDF-ABLAGE
WRITE N = UMBENENNEN
WRITE W = FENSTER
WRITE ! = ENDE
WRITE # = LOESCHEN
WRITE C = GRUEN UMFAERBEN
WRITE R = DREHEN
WRITE S = ZOLL => CM UMRECHNEN
WRITE Z = LUPE
READ &V
IF &V=&STR(!) THEN DO
    ENDGS
    GOTO FIN
    QUIT
IF &V=&STR(B) THEN DO
    ENDGS
    GOTO BI
    QUIT
IF &V=&STR(C) | &V=&STR(R) | &V=&STR(Z) | &V=&STR(S) | +
&V=&STR(N) THEN GOTO CH
IF &V=&STR(#) THEN DO
    DPIB &BIN B1
    WRITE DAS BILD &BIN WURDE AUS DEM BUCH &GSB AUSTRADERT.

```

92/03/25

```

ENDGS
GOTO BI
QUIT
IF &V=&STR(G) THEN DO
  CW &AP
  OW GDF
  ROUT
  CW GDF
  OW &AP
  WRITE DAS BILD &BIN WURDE INS BUCH &ZUB. EINGETRAGEN
  QUIT
IF &V=&STR(W) THEN DO
  WRITE FENSTERWAHL / &FORM / =:
  WRITENR ( "H" = DIN-H | "Q" = DIN-Q | SONST MAXIMAL )
  READ &ANS
  IF &ANS=&STR() THEN SET &FORM=&ANS
  IF &FORM=&STR(H) && &FORM=&STR(Q) THEN GOTO IK
  SET &EFE=&STR(A4)&FORM
  IF &XLU < 0 THEN SET &XLU = 0
  IF &YLU < 0 THEN SET &YLU = 0
  DWI &XLU &YLU &EFE
  IK: ROUT
  QUIT
GOTO MP
CH: IF &V=&STR(C) THEN SAT C2
IF &V=&STR(R) THEN DO
  WRITENR DREHWINKEL ( IN GRAD ) =:
  READ &ANS
  SET &WI = &STR(&ANS)
  ROT 0 0 &WI
  QUIT
IF &V=&STR(N) THEN DO
  SET &LB=&LENGTH(&BIN)
  ER: WRITENR NEUE NAME / &BIN / =: ( "ENTER" = KEINE NEUE NAME )
  READ &ANS
  IF &ANS=&STR() THEN GOTO MP
  IF &LENGTH(&ANS) > &LB THEN DO
    WRITE NAMENSLAENGE > &LB ZEICHEN !
    GOTO ER
  QUIT
  SET &NEU=&ANS
  WRITENR NEUE NAME &NEU IN ORDNUNG ? ( "ENTER" = JA )
  READ &ANS
  IF &ANS=&STR() THEN GOTO ER
  REN &BIN &NEU
  SET &BIN=&NEU
  QUIT
IF &V = &STR(S) THEN DO
  PWI
  SCA 0. 0. 1. 2.54 1. 2.54
  ROUT
  QUIT
IF &V=&STR(Z) THEN DO
  LU: WRITENR ECKE LINKSUNTEN : X / &XLU / =:, Y / &YLU / =:
  READ &AX &AY
  IF &STR(&AX) = &STR() THEN SET &XLU = &STR(&AX)
  IF &STR(&AY) = &STR() THEN SET &YLU = &STR(&AY)
  WRITE ECKE LINKSUNTEN = ( &XLU , &YLU )
  WRITENR ECKE RECHTSOBEN : X / &XRO / =:, Y / &YRO / =:

```

```

READ &AX &AY
IF &STR(&AX) ^= &STR() THEN SET &XRO = &STR(&AX)
IF &STR(&AY) ^= &STR() THEN SET &YRO = &STR(&AY)
WRITE ECKE RECHTSOBEN = ( &XRO , &YRO )
API &BIN
PWI &XLU &YLU &XRO &YRO
QUIT
WRITE GEAEANDERTES &BIN ABSPEICHERN ? ( # = JA ) =:
READ &ANS
IF &ANS=&STR(#) THEN DO
    SPIB &BIN B1
    WRITE DAS BILD &BIN WURDE IN &GSB. GESPEICHERT
    QUIT
GOTO MP
FIN: SET &KAMEL=&STR(&BIN &FORM &XLU &YLU &XRO &YRO)
EX '&SYSUID..CLIST(BEKI)' 'MEM(IMMANI)'
SET &KAMEL=&GOBLIN
EX '&SYSUID..CLIST(ALOGS)'
EXIT

```

PICOR

PICOR überträgt die Strukturliste des GS-Bildes BILD in die gegliederte Datei 'inr000.PICOR(BILD)'.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(GLIEDER)
- inr000.CLIST(NEVE)
- inr000.CLIST(TAGOK)

Benötigte Hilfsdatei :

- inr000.MEMORY(PICOR)

Die Prozedurliste:

```

PROC 0 SIZE(999)
CONTROL PROMPT MAIN MSG.END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 BIN TON
SET &SED=&SYSUID..PICOR
IF &SYSDSN('&SED') ^= OK THEN +
EX '&SYSUID..CLIST(ADA)' '&SED LIN(80) KA(10)'
WRITE
WRITE GS-PROZEDUR PICOR
WRITE STRUKTUR EINES GS-BILDES WIRD IN DIE DATEI '&SED.' +
GESPEICHERT.
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(PICOR)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &BIN
SET &KAMEL=&STR()
WRITE GS-BUCH:
SET &N3=&STR(#)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
IF &N1 = &STR(#) THEN GOTO FIE
WRITE BEGINN DER PROZEDUR

```

91/02/22

```

ALLOC FI(B1) DS('&TON') SHR REU
BI: WRITE NAME DES BILDES :
SET &LAB=&BIN
EX '&SYSUID..CLIST(TAGOK)'
IF &BIN=&STR(!) THEN DO
    SET &BIN=&LAB
    GOTO FIN
    QUIT
WRITENR BILD &BIN O.K. =: ( "ENTER" : JA )
READ &AN
IF &AN = &STR() THEN GOTO BI
SET &SEM=&SED(&BIN.)
ALLOC FILE(INTEX) DA('&SEM') SHR REU
SET &SYSOUTTRAP = 500
INITGS -&SIZE
RPIB &BIN B1
LS
OPENFILE INTEX OUTPUT
SET I=2
ERROR
DO WHILE &I <= &SYSOUTLINE
    SET LIN = &STR(&&SYSOUTLINE&I)
    SET INTEX = &STR(&LIN)
    PUTFILE INTEX
    SET &I = &I+1
    QUIT
CLOSFILE INTEX
ENDGS
SET &SYSOUTTRAP = 0
SET &I=&I-6
WRITE DIE STRUKTUR DES BILDES &BIN ( &I. ZEILEN ) WURDE
WRITE IN DER DATEI '&SEM' GESPEICHERT.
GOTO BI
FIN: SET &KAMEL = &KAMEL &BIN
EX '&SYSUID..CLIST(BEKI)' 'MEM(PICOR)'
FREE FI(B1 INTEX)
WRITENR COMPRESS '&SED' ? =: ( "ENTER" = JA )
READ &ANS
IF &ANS=&STR() THEN DO
    COMPRESS '&SED' NOLIST
    REDUCE '&SED'
    QUIT
FIE: EXIT

```

PIXI

PIXI vereinfacht Eröffnung und Schließung einer GS-Sitzung.
Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(ALOGS)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)
- inr000.CLIST(TUKOR)

Benötigte Hilfsdatei :

- inr000.MEMORY(BOOKS)

Die Prozedurliste:

```
PROC 0
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON AP
WRITE GS-PROZEDUR      PIXI                      91/02/20
WRITE EINSTIEG ZUM ZEICHNEN MIT DEM GS .
SET &EA=0
EX '&SYSUID..CLIST(ALOGS)'
SET &SYSDVAL=&KAMEL
READDVAL &SIZE
IF &SIZE=&STR(0) THEN GOTO FIN
WRITE BEGINN DER PROZEDUR
WRITE                                     ****
WRITE DIE PROZEDUR KANN MAN DURCH EINTIPPEN VON STOP BEENDEN.
WRITE                                     ****
WRITE SIZE = &STR(-&SIZE)
INITGS -&SIZE
OW &AP
DWI MAX
TERMIN STOP
ENDGS
EX '&SYSUID..CLIST(ALOGS)'
FIN: EXIT
```

PODEL

PODEL unterstützt den Benutzer beim ausradiieren einzelner Elemente aus einem komplexen Objekts eines GS-Bildes.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(ALOGS)
- inr000.CLIST(BEKI)
- inr000.CLIST(GLIEDER)
- inr000.CLIST(NEVE)
- inr000.CLIST(OBJEKT)
- inr000.CLIST(TAGOK)
- inr000.CLIST(TUKOR)

Benötigte Hilfsdateien :

- inr000.MEMORY(PONUM)
- inr000.MEMORY(BOOKS)

Die Prozedurliste:

```
PROC 0
CONTROL PROMPT MAIN MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 BIN TON AP
WRITE
WRITE GS-PROZEDUR PODEL                          91/02/27
WRITE PRIMITIVE OBJEKTE EINES COMPLEXEN OBJEKTS
WRITE WERDEN AUSRADIERT
SET &EA=0
EX (ALOGS)
SET &GOBLIN=&KAMEL
```

```

SET &SYSDVAL=&KAMEL
READDVAL &SIZE &GSB &UFB &SF &ZUB
IF &SIZE=&STR(0) THEN EXIT
SET &EA=0
EX (BEKI) 'MEM(PONUM)'
SET &SYSDVAL=&KAMEL
READDVAL &BIN &PAN &JI &JF &FORM &MG &MH &XLU &YLU &XRO &YRO
SET &N3=&PAN
SET &TON=&GSB
WRITE BEGINN DER PROZEDUR
SET &KAMEL=&FORM &MG &MH &XLU &YLU &XRO &YRO
NBO: EX (OBJEKT)
SET &PAN=&N3
SET &SYSDVAL=&KAMEL
READDVAL &FORM &MG &MH &XLU &YLU &XRO &YRO
SET &EFE=&STR(A4)&FORM
INITGS -&SIZE
OW &AP
DWI 0 0 &EFE
RPIB &BIN B1
PWI &XLU &YLU &XRO &YRO
IF &PAN != &STR(#) THEN APA &PAN
APO
IR: SET &V = &STR()
WRITE
WRITE VORGEHEN =:   W : BILD ZEIGEN
WRITE              ! : ENDE
WRITE              0 : OBJEKT / FENSTER AENDERN
WRITE              R : UNKONTROLLIERT RADIEREN
WRITE              SONST : KONTROLLIERT RADIEREN
READ &V
IF &V=&STR(!) THEN GOTO FGS
IF &V=&STR(W) THEN GOTO AHA
IF &V=&STR(0) THEN GOTO FGS
EL: WRITE ELEMENTE VON / &JI / BIS / &JF / =: =: ( < 1 : ENDE )
READ &A1 &A2
WRITE A1=&A1 A2=&A2
IF &A1 = &STR(0) THEN DO
  APO
  GOTO IR
  QUIT
IF &A1 != &STR() THEN SET &JI=&A1
IF &A2 = &STR() | &A2 < &A1 THEN SET &JF=&A1
ELSE SET &JF=&A2
WRITE JI = &JI , JF = &JF
IF &V = &STR(R) THEN GOTO RB
APO &JI &JF
SAT C&MH
ROUT
WRITE WEGRADIEREN ? =: ( # : JA )
READ &ANS
IF &ANS != &STR(#) THEN GOTO EL
RB: DPO &JI &JF
WRITE PRIMITIVE OBJEKTE &JI - &JF BESEITIGT
GOTO EL
AHA: LA
ROUT
WRITE GEAENDERTES &BIN ABSPEICHERN ? ( # = JA ) =:
READ &ANS

```

```

IF &ANS=&STR(#) THEN DO
  PWI
  SPIB &BIN B1
  WRITE DAS BILD &BIN WURDE IN &GSB. GESPEICHERT
  QUIT
GOTO IR
FGS: ENDGS
IF &V = &STR(!) THEN GOTO NBO
FIN: SET &KAMEL=&GOBLIN
EX (ALOGS)
SET &KAMEL=&BIN &PAN &JI &JF &FORM &MG &MH &XLU &YLU &XRO &YRO
EX (BEKI) 'MEM(PONUM)'
EXIT

```

PONUM

PONUM erleichtert die Identifikation einzelner Elemente in einem komplexen Objekt eines GS-Bildes durch umfärben der Elemente.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(ALOGS)
- inr000.CLIST(BEKI)
- inr000.CLIST(GLIEDER)
- inr000.CLIST(NEVE)
- inr000.CLIST(OBJEKT)
- inr000.CLIST(TAGOK)
- inr000.CLIST(TUKOR)

Benötigte Hilfsdateien :

- inr000.MEMORY(PONUM)
- inr000.MEMORY(BOOKS)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 BIN TON AP
WRITE
WRITE GS-PROZEDUR PONUM
WRITE ELEMENTE EINES BILDTTEILS
WRITE WERDEN UMGEFÄRBT
WRITE UM IHREN ORDNUNGSZAHLEN ZU FINDEN.
SET &EA=0
EX (ALOGS)
SET &GOBLIN=&KAMEL
SET &SYSDVAL=&KAMEL
READDVAL &SIZE &GSB &UFB &SF &ZUB
IF &SIZE=&STR(0) THEN EXIT
SET &EA=0
EX (BEKI) 'MEM(PONUM)'
SET &NOTNAGL=&KAMEL
SET &SYSDVAL=&KAMEL
READDVAL &BIN &PAN &JI &JF &FORM &MG &MH &XLU &YLU &XRO &YRO
SET &N3=&PAN
SET &TON=&GSB
WRITE BEGINN DER PROZEDUR

```

91/02/27


```

SET &KAMEL=&FORM &MG &MH &XLU &YLU &XRO &YRO
NBO: EX (OBJEKT)
SET &PAN=&N3
SET &SYSDVAL=&KAMEL
READDVAL &FORM &MG &MH &XLU &YLU &XRO &YRO
SET &EFE=&STR(A4)&FORM
INITGS -&SIZE
OW &AP
DWI 0 0 &EFE
RPIB &BIN B1
PWI &XLU &YLU &XRO &YRO
IF &PAN ≠ &STR(#) THEN APA &PAN
APO
SAT C&MG
JOC: SET &V = &STR()
WRITE
WRITE VORGEHEN =:   W : BILD ZEIGEN
WRITE                ! : ENDE
WRITE                0 : OBJEKT / FENSTER AENDERN
WRITE                E : EINZELELEMENTE UMFAERBEN
WRITE                SONST : REIHENWEISE UMFAERBEN
READ &V
IF &V = &STR(!) | &V = &STR(0) THEN GOTO FGS
IF &V = &STR(W) THEN GOTO AHA
IF &V = &STR(E) THEN GOTO EL
WRITE ANFANG / &JI / =:
READ &AI
IF &AI ≠ &STR() THEN SET &JI=&AI
IF &JI < 1 THEN SET &JI = 1
WRITE ENDE / &JF / =:
READ &AF
IF &AF ≠ &STR() THEN SET &JF=&AF
IF &JF < &JI THEN SET &JF = &JI + 1
WRITE ABSTAND / &JD / =:
READ &AD
IF &AD ≠ &STR() THEN SET &JD=&AD
IF &JD < 1 THEN SET &JD = 1
SET &K=0
SET &LF=&JI-1
SET &M=-1
DO WHILE K < 49
  SET &M=&M+1
  IF &M ≥ 7 THEN SET &M = &M-7
  SET &K=&K+1
  SET &LI=&LF+1
  SET &LF=&LI+&JD-1
  IF &LF > &JF THEN SET &LF = &JF
  WRITE &LI &STR( < L < ) &LF &STR( , ) C&M
  APO &LI &LF
  SAT C&M
  IF &LF = &JF THEN GOTO CR
  QUIT
CR: ROUT
  APO &JI &JF
  SAT C&MG
  APO
GOTO JOC
EL: WRITE ELEMENTNUMMER / &JI / =: ( < 1 : ENDE )
READ &ANS

```

```

IF &ANS = &STR() THEN GOTO N8
IF &ANS < 1 THEN DO
    ROUT
    APO
    GOTO JOC
    QUIT
IF &ANS ^= &STR() THEN SET &JI=&ANS
N8: APO &JI
SAT C&MH
GOTO EL
AHA: LA
ROUT
WRITE GEANDERTES &BIN ABSPEICHERN ? ( # = JA ) =:
READ &ANS
IF &ANS=&STR(#) THEN DO
    PWI
    SPIB &BIN B1
    WRITE DAS BILD &BIN WURDE IN &GSB. GESPEICHERT
    QUIT
GOTO JOC
FGS: ENDGS
IF &V ^= &STR(!) THEN GOTO NBO
FIN: SET &KAMEL=&GOBLIN
EX (ALOGS)
SET &KAMEL=&BIN &PAN &JI &JF &FORM &MG &MH &XLU &YLU &XRO &YRO
EX (BEKI) 'MEM(PONUM)'
EXIT

```

Die im Folgenden beschriebene Prozeduren stimmen mit dem im SERVUS-Handbuch /6/ abgedruckten Prozeduren überein.

DATAS

Mit DATAS kann man SERVUS-Dateien im Vordergrund lesen und kopieren.
Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)

Benötigte Hilfsdatei :

- inr000.MEMORY(DATAS)

Die Prozedurliste:

```
PROC 0 IR1(00) IR2(00) IR3(00) IW(00)
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE
WRITE PROZEDUR      DATAS
WRITE EINE SERVUS-DATEI WIRD GESICHTET ( UND KOPIERT )
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(DATAS)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &PR2 &D2 &PR3 &D3 &PRR &DR
SET &KAMEL=&STR()
SET &N3 = &STR(DATA)
WRITE LESEDATEI NR "15" :
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
SET &DAT1=&TON
IF &DAT1 = &STR() THEN GOTO FIN
SET &IR1=&STR(15)
WRITE LESEDATEI NR "16" :
SET &N1=&PR2
SET &N2=&D2
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
SET &DAT2=&TON
IF &DAT2 = &STR() THEN DO
    SET &KAMEL = &KAMEL &PR3 &D3
    SET &DAT3=&STR()
    GOTO WRI
END
SET &IR2=&STR(16)
WRITE LESEDATEI NR "17" :
SET &N1=&PR3
SET &N2=&D3
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
SET &DAT3=&TON
IF &DAT3 = &STR() THEN SET &IR3=&STR(17)
WRI: WRITE SCHREIB-DATEI "26" :
```

91/09/30

```

SET &N1=&PRR
SET &N2=&DR
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
SET &DATR=&TON
IF &DATR ^= &STR() THEN SET &IW=&STR(26)
BEG: EX '&SYSUID..CLIST(BEKI)' 'MEM(DATAS)'
WRITE BEGINN DER PROZEDUR
ALLOC DA('&DAT1') F(FT15F001) SHR REU
IF &DAT2 ^= &STR() THEN ALLOC DA('&DAT2') F(FT16F001) SHR REU
IF &DAT3 ^= &STR() THEN ALLOC DA('&DAT3') F(FT17F001) SHR REU
IF &DATR ^= &STR() THEN ALLOC DA('&DATR') F(FT26F001) SHR REU
SET &SYSDVAL=&STR(&IR1 &IR2 &IR3 &IW)
CALL 'INR105.SERVUS.LOAD(DATAS)' '&SYSDVAL'
IF &DAT2 ^= &STR() THEN FREE DA('&DAT2')
IF &DAT3 ^= &STR() THEN FREE DA('&DAT3')
IF &DATR ^= &STR() THEN FREE DA('&DATR')
FREE DA('&DAT1')
FIN: WRITE PROZEDUR DATAS ABGESCHLOSSEN.
EXIT

```

DDFIG

DDFIG hilft - im Rahmen eines Dialogs - dem Benutzer, geeignete Funktionen einer SERVUS-Datei auszuwählen und sie als Funktionsoberfläche abzubilden. Die SERVUS-Datei muß "GRA4" normiert sein.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)
- inr000.CLIST(SERVIN)
- inr000.CLIST(TUKOR)

Benötigte Hilfsdateien :

- inr000.MEMORY(FIGUR)
- inr000.memory(laddfi)
- inr000.MEMORY(tttt)
- inr000.MEMORY(eeee)
- inr000.MEMORY(xxxx)
- inr000.MEMORY(tttt)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON AP
WRITE
WRITE PROZEDUR          DDFIG
WRITE DARSTELLUNG VON FUNKTIONEN ALS KURVENSCHAREN.
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(FIGUR)'
SET &SYSDVAL=&KAMEL
READDVAL &AP &N1 &N2 &PRB &BH
EX '&SYSUID..CLIST(TUKOR)'
SET &KAMEL=&AP
WRITE FUNKTIONS-DATEI :

```

91/09/30

```

SET &N3 = &STR(DATA)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &DAT1=&TON
WRITE "BUCH" ZUM BILDSPEICHERN :
SET &N1 = &PRB
SET &N2 = &BH
SET &N3 = &STR(#)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1) CQ(1)'
SET &BUCH=&TON
EX '&SYSUID..CLIST(BEKI)' 'MEM(FIGUR)'
EX '&SYSUID..CLIST(SERVIN)' 'DDFIG'
IF &N1 = &STR(#) THEN GOTO FIN
WRITE BEGINN DER PROZEDUR
IF &AP=&STR(PCG) THEN +
ALLOC F(ADMPC) DA('TSOSYS.GDDM.GDDMSAM') SHR
ALLOC DA('&DAT1') F(FT15F001) SHR REU
IF &BUCH = &STR() THEN GOTO PI
IF &SYSDSN('&BUCH')=>OK THEN DO
    EX '&SYSUID..CLIST(ADA)' '&BUCH LIN(0) KA(10)'
    WRITE BUCH '&BUCH' ALLOKIERT.
    END
ALLOC DA('&BUCH') F(TRACEG7) SHR REU
PI: SET &SYSDVAL = &STR(&AP &KAMEL &N1)
CALL 'INR105.SERVUS.LOAD(DDFIG)' '&SYSDVAL'
FREE DA('&DAT1')
IF &AP=&STR(PCG) THEN FREE F(ADMPC)
FIN: WRITE PROZEDUR DDFIG BEENDET.
EXIT

```

DRUDA

Die Prozedur stellt die Steuerkarten eines Druck-Auftrages - im Dialog mit dem Benutzer - zusammen.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)
- inr000.CLIST(FEJ)
- inr000.CLIST(SUSA)

Benötigte Hilfsdatei :

- inr000.MEMORY(DRUDA)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE
WRITE PROZEDUR DRUDA
WRITE FUNKTIONEN EINER SERVUS DATEI WERDEN AUSGEDRUCKT.
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(DRUDA)'
SET &SYSDVAL=&STR(&KAMEL)

```

91/02/21

```

READDVAL &N1 &N2 &N3 &INL &KE &KL
SET &KAMEL=&STR()
WRITE DATEI :
EX '&SYSUID..CLIST(NEVE)'
IF &N1 = &STR(#) THEN GOTO FIN
WRITE KURVENZAHL / &INL / =: ( < 51 )
WRITE > 0 : KURVEN EINZELN EINGEBEN ,
WRITE = 0 : KURVEN VOM ERSTEN BIS LETZTEN
WRITE < 0 : ERSTE , ERSTE + ABSTAND , ... , +
ERSTE + (ABS(&INL)-1)*ABSTAND
READ &ANS
IF &STR(&ANS)=-&STR() THEN SET &INL=&ANS
IF &INL <= 0 THEN DO
  WRITENR ERSTE KURVE / &KE / =:
  READ &ANS
  IF &STR(&ANS)=-&STR() THEN SET &KE=&ANS
  IF &INL = 0 THEN WRITENR LETZTE KURVE / &KL / =:
  IF &INL < 0 THEN WRITENR KURVENABSTAND / &KL / =:
  READ &ANS
  IF &STR(&ANS)=-&STR() THEN SET &KL=&ANS
  QUIT
SET &KAMEL=&STR(&KAMEL &INL &KE &KL )
EX '&SYSUID..CLIST(BEKI)' 'MEM(DRUDA)'
IF &INL > 0 THEN DO
  SET &L=0
  SET &K=0
  SET &B=0
  DO WHILE &K < 6
    SET &K=&K+1
    SET &LU=&STR()
    SET &KB=&K*10
    DO WHILE &L<&KB
      IF &L >= &INL THEN GOTO EDI
      SET &L=&L+1
      S1: WRITENR &L-TE KURVE =: +
      ( "ENTER"= ENDE | 0 = NEUES BLATT )
      READ &NU
      IF &STR(&NU) = &STR() THEN GOTO EDI
      IF &STR(&NU) = &STR(0) THEN DO
        SET &B=0
        GOTO S1
      QUIT
      IF &B > 0 && &SUBSTR(1,&NU) =- &STR(-) THEN +
      SET &NU=&STR(-&STR(&NU))
      IF &STR(&LU)=&STR() THEN SET &LU=&STR(&NU)
      ELSE SET &LU=&STR(&LU,&STR(&NU))
      SET &B=&B+1
      SET &LAUF&K=&STR(&LU)
      QUIT
    QUIT
  EDI: IF &STR(&NU) = &STR() THEN SET &L=&L-1
  QUIT
WRITE BEARBEITUNG DES AUFTRAGES :
EX '&SYSUID..CLIST(FEJ)'
EDIT '&SYSUID..&N3..CNTL' OLD CNTL NUM
0120 //DRUCK EXEC F7CLG,USER='INR105.SERVUS.LOAD'
0130 //C.SYSRINT DD DUMMY
0140 //C.SYSIN DD *
0150 PROGRAM DRUDA

```

```

0160      CALL PAGEFF
0170      STOP
0180      END
0190     /**
0191     /**/L.SYSPRINT DD DUMMY
0200     /**/L.SYSIN DD *
0210     INCLUDE SYSLIB(SERDIO,COPYDA,NEXT)
0220     ENTRY DRUDA
0230     /**
0300     /**G.FT11F001 DD DSN=&TON,DISP=SHR
0320     /**G.SYSPRINT DD SYSOUT=*
0330     /**G.SYSIN DD *
0400     &&LADRUC IRE=11,KPRI=0,MODPRI='DN.EINX1',
IF &INL > 0 THEN DO
    0421     LAUF=&LAUF1,
    IF &K > 1 THEN 0422     &LAUF2,
    IF &K > 2 THEN 0423     &LAUF3,
    IF &K > 3 THEN 0424     &LAUF4,
    IF &K > 4 THEN 0425     &LAUF5,
    IF &K > 5 THEN 0425     &LAUF6,
    0429     INLIST=&L, &&END
    QUIT
IF &INL <= 0 THEN +
    0421     LAUF=&KE,&KL,INLIST=&INL, &&END
1999     &&LADRUC IRE=0, &&END
2000     /**
EX '&SYSUID..CLIST(SUSA)''&N3.'
FIN: EXIT

```

ENDE

Die Prozedur schreibt eine Endzeile in eine SERVUS-Datei.
Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)

Benötigte Hilfsdatei :

- inr000.MEMORY(DATAS)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE
WRITE PROZEDUR      ENDE
WRITE EINE SERVUS-DATEI WIRD MIT EINER ENDZEILE VERSEHEN
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(DATAS)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &PR2 &D2 &PR3 &D3 &PRR &DR
SET &KAMEL=&STR()
SET &N3 = &STR(DATA)
WRITE DATEI :

```

91/09/30

```

EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &KAMEL=&KAMEL &PR2 &D2 &PR3 &D3 &PRR &DR
BEG: EX '&SYSUID..CLIST(BEKI)' 'MEM(DATAS)'
WRITE BEGINN DER PROZEDUR
ALLOC DA('&TON') F(FT15F001) SHR REU
CALL 'INR105.SERVUS.LOAD(ENDE)'
FREE DS('&TON')
WRITE ENDE DER PROZEDUR "ENDE"
FIN: EXIT

```

FIGUR

FIGUR hilft - im Rahmen eines Dialogs - dem Benutzer, geeignete Funktionen einer SERVUS-Datei auszuwählen und sie als Kurvenschar abzubilden. Die SERVUS-Datei muß "GRA4" normiert sein.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)
- inr000.CLIST(SERVIN)
- inr000.CLIST(TUKOR)

Benötigte Hilfsdateien :

- inr000.MEMORY(FIGUR)
- inr000.memory(lafigu)
- inr000.MEMORY(tttt)
- inr000.MEMORY(eeee)
- inr000.MEMORY(xxxx)
- inr000.MEMORY(tttt)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON AP
WRITE
WRITE PROZEDUR          FIGUR
WRITE DARSTELLUNG VON FUNKTIONEN ALS KURVENSCHAREN.
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(FIGUR)'
SET &SYSDVAL=&KAMEL
READDVAL &AP &N1 &N2 &PRB &BH
EX '&SYSUID..CLIST(TUKOR)'
SET &KAMEL=&AP
WRITE FUNKTIONS-DATEI :
SET &N3 = &STR(DATA)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &DAT1=&TON
WRITE "BUCH" ZUM BILDSPEICHERN :
SET &N1 = &PRB
SET &N2 = &BH
SET &N3 = &STR(#)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1) CQ(1)'

```

91/02/21


```

SET &BUCH=&TON
EX '&SYSUID..CLIST(BEKI)' 'MEM(FIGUR)'
EX '&SYSUID..CLIST(SERVIN)' 'FIGUR'
IF &N1 = &STR(#) THEN GOTO FIN
WRITE BEGINN DER PROZEDUR
IF &AP=&STR(PCG) THEN +
ALLOC F(ADMPC) DA('TSOSYS.GDDM.GDDMSAM') SHR
ALLOC DA('&DAT1') F(FT15F001) SHR REU
IF &BUCH = &STR() THEN GOTO PI
IF &SYSDSN('&BUCH')=&OK THEN DO
    EX '&SYSUID..CLIST(ADA)' '&BUCH LIN(0).KA(10)'
    WRITE BUCH '&BUCH' ALLOKIERT.
    END
ALLOC DA('&BUCH') F(TRACEGS7) SHR REU
PI: SET &SYSDVAL = &STR(&AP &KAMEL &N1)
CALL 'INR105.SERVUS.LOAD(FIGUR)' '&SYSDVAL'
IF &AP=&STR(PCG) THEN FREE F(ADMPC)
FIN: WRITE PROZEDUR FIGUR BEENDET.
EXIT

```

FUNKIN

FUNKIN hilft - im Rahmen eines Dialogs - dem Benutzer, eine Funktion zusammenzustellen und in eine SERVUS-Datei einzutragen.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)

Benötigte Hilfsdatei :

- inr000.MEMORY(DATAS)

Die Prozedurliste:

```

PROC 0 IWR(16) IRE(15)
CONTROL MAIN MSG PROMPT END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE
WRITE PROZEDUR      FUNKIN                      91/02/22
WRITE EINE FUNKTION WIRD IN EINE SERVUS DATEI GESCHRIEBEN .
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(DATAS)'
SET &SYSDVAL=&KAMEL
READDVAL &PR &DR &PR2 &D2 &PR3 &D3 &PW &DW
SET &KAMEL=&STR()
SET &N3=&STR(DATA)
WRITE LESEDATEI :
SET &N1=&PR
SET &N2=&DR
EX '&SYSUID..CLIST(NEVE)' 'LQ(1) CQ(0)'
IF &N1 = &STR(#) | &IRE = 0 THEN DO
    SET &N2=&STR(*?#!@~|%)
    SET &IRE=&STR(00)
    QUIT
ELSE DO

```

```

    IF &LENGTH(&STR(&IRE)) < 2 THEN SET &IRE=&STR(0&IRE)
    SET &FRE=&STR(FT)&STR(&IRE)&STR(F001)
    ALLOC DA('&TON') F(&FRE) SHR REU
    QUIT
SET &PR=&N1
SET &DR=&STR(&N2)
SET &KAMEL=&KAMEL &STR(&PR2 &D2 &PR3 &D3)
WRITE SCHREIBDATEI :
IF &LENGTH(&STR(&IWR)) < 2 THEN SET &IWR=&STR(0&IWR)
SET &FWR=&STR(FT)&STR(&IWR)&STR(F001)
SET &N1=&PW
SET &N2=&DW
EX '&SYSUID..CLIST(NEVE)' 'LQ(1) CQ(0)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &PW=&N1
SET &DW=&N2
ALLOC DA('&TON') F(&FWR) SHR REU
BEG: EX '&SYSUID..CLIST(BEKI)' 'MEM(DATAS)'
WRITE BEGINN DER PROZEDUR
SET &SYSDVAL = &STR(&IRE &IWR &PW &DW &PR &DR)
CALL 'INR105.SERVUS.LOAD(FUNKIN)' '&SYSDVAL '
WRITE ENDE DER PROZEDUR FUNKIN
FIN: EXIT

```

IDA

Prozedur zum anlegen einer SERVUS-Datei.
Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(NEVE)

Benötigte Hilfsdateien :

- keine

Die Prozedurliste:

```

PROC 0 USER(&SYSUID.) DD(TEST) NORM(FOL8)
CONTROL MAIN MSG PROMPT
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE
WRITE PROZEDUR      IDA
WRITE EINE SERVUS DATEI WIRD ANGELEGT | RENORMIERT
WRITE
WRITE ANGABEN ZU DER DATEI :
SET &N1=&USER
SET &N2=Dampfdruck
SET &N3=&STR(DATA)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1) CQ(1) SQ(1)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &USER=&N1
IF &SYSDSN('&TON')=&OK THEN EX '&SYSUID..CLIST(ADA)' '&TON LIN(X)'
WRITE DIE DATEI IST ANGELEGT . (RE-)NORMIERUNG DER DATEI:
WRITENR NORM / &NORM / =: +
( "ENTER"(=JA) | GRA4 | GRA8 | FOL8 | PLOT )
READ &ANS
IF &ANS=&STR() THEN SET &NORM=&ANS

```

91/02/22

```

SET &SYSDVAL = &N2 &NORM &USER &VOL
ALLOC DA('&TON') F(FT15F001) SHR REU
CALL 'INR105.SERVUS.LOAD(IDAJOB)' '&SYSDVAL'
WRITE ENDE DER PROZEDUR IDA
EXIT

```

KOPIER

Die Prozedur stellt die Steuerkarten eines Kopier-Auftrages - im Dialog mit dem Benutzer - zusammen .

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(FEJ)
- inr000.CLIST(NEVE)
- inr000.CLIST(SUSA)

Benötigte Hilfsdatei :

- inr000.MEMORY(KOPIER)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE
WRITE PROZEDUR      KOPIER
WRITE EINE SERVUS DATEI WIRD GELESEN ( UND KOPIERT ) .
WRITE
SET &EA=0
EX (BEKI) 'MEM(KOPIER)'
SET &SYSDVAL=&STR(&KAMEL)
READDVAL &N1 &N2 &N3 &PROW &WRDAT &INL &KE &KL
SET &KAMEL=&STR()
WRITE LESEDATEI :
WRITENR MIT KOPF ( "ENTER"=JA ) =:
READ &ANS
IF &ANS=&STR() THEN SET &URE=&STR(-11)
ELSE SET &URE=11
EX (NEVE)
IF &N1 = &STR(#) THEN GOTO FIN
SET &DARE=&TON
WRITE SCHREIBDATEI :
SET &N1=&PROW
SET &N2=&WRDAT
SET &N3=&STR(DATA)
EX (NEVE) 'LQ(1)'
SET &DAWR=&TON
IF &DAWR=&STR() THEN DO
    SET &UWR=0
    SET &PP=2
    QUIT
ELSE DO
    SET &UWR=15
    SET &PP=1
    QUIT

```

91/02/27

```

WRITE KURVENZAHL / &INL / =: ( < 51 )
WRITE > 0 : KURVEN EINZELN EINGEBEN ,
WRITE = 0 : KURVEN VOM ERSTEN BIS LETZTEN
WRITE < 0 : ERSTE , ERSTE + ABSTAND , ... , +
ERSTE + (ABS(&INL)-1)*ABSTAND
READ &ANS
IF &STR(&ANS)≠&STR() THEN SET &INL=&ANS
IF &INL <= 0 THEN DO
  WRITENR ERSTE KURVE / &KE / =:
  READ &ANS
  IF &STR(&ANS)≠&STR() THEN SET &KE=&ANS
  IF &INL = 0 THEN WRITENR LETZTE KURVE / &KL / =:
  IF &INL < 0 THEN WRITENR KURVENABSTAND / &KL / =:
  READ &ANS
  IF &STR(&ANS)≠&STR() THEN SET &KL=&ANS
  QUIT
SET &KAMEL=&STR(&KAMEL &INL &KE &KL &DR)
EX (BEKI) 'MEM(KOPIER)'
IF &INL > 0 THEN DO
  SET &L=0
  SET &K=0
  DO WHILE &K < 5
    SET &K=&K+1
    SET &KB=&K*10
    SET &LU=&STR()
    DO WHILE &L<&KB
      IF &L >= &INL THEN GOTO EDI
      SET &L=&L+1
      WRITENR &L-TE KURVE =: ( "ENTER"= ENDE )
      READ &NU
      IF &NU = &STR() THEN GOTO EDI
      SET &LU=&LU&STR(,)&NU
      SET &LAUF&K=&LU
    QUIT
  QUIT
  EDI: IF &NU = &STR() THEN SET &L=&L-1
  QUIT
WRITE BEARBEITUNG DES AUFTRAGES :
EX (FEJ)
EDIT &N3..CNTL OLD NUM
0120 //COPY EXEC F7CLG,USER='INR105.SERVUS.LOAD'
0130 //C.SYSPRINT DD DUMMY
0140 //C.SYSIN DD *
0150     PROGRAM KOPIER
0160     CALL COPYDA
0170     STOP
0180     END
0190 //*
0191 //L.SYSPRINT DD DUMMY
0200 //L.SYSIN DD *
0210 INCLUDE SYSLIB(SERDIO,COPYDA,FENST4,NEXT)
0220 ENTRY KOPIER
0230 //*
0300 //G.FT11F001 DD DSN=&DARE,DISP=SHR
IF &UWR ≠ 0 THEN +
0310 //G.FT15F001 DD DSN=&DAWR,DISP=SHR
0320 //G.SYSPRINT DD SYSOUT=*
0330 //G.SYSIN DD *
0400 &&LAKOPI IRE=&URE,IWR=&UWR,MODUS='**OLD**',KPRI=&PP,

```

```

0410 LAX=0,XMIST=0.,XMAST=0.,XEDST=0.,
0420 LAF=0,FMIST=0.,FMAST=0.,FEDST=0.,
IF &INL > 0 THEN DO
  0421 LAUF=&LAUF1,
  IF &L > 10 THEN 0422 &LAUF2,
  IF &L > 20 THEN 0423 &LAUF3,
  IF &L > 30 THEN 0424 &LAUF4,
  IF &L > 40 THEN 0425 &LAUF5,
  0429 INLIST=&L, &&END
QUIT
IF &INL <= 0 THEN +
  0421 LAUF=&KE,&KL,INLIST=&INL, &&END
1999 &&LAKOPI IRE=0, &&END
2000 //
EX (SUSA) '&N3.'
FIN: EXIT

```

LOESCH

Die Prozedur stellt die Steuerkarten eines Loesch-Auftrages - im Dialog mit dem Benutzer - zusammen. Der Benutzer erhält dabei Gelegenheit, eine Sicherheitskopie der zu bearbeitenden SERVUS-Datei im Massenspeicher abzulegen.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(FEJ)
- inr000.CLIST(NEVE)
- inr000.CLIST(SUSA)

Benötigte Hilfsdatei :

- inr000.MEMORY(LOESCH)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE
WRITE PROZEDUR      LOESCH
WRITE KURVEN AUS EINER SERVUS DATEI WERDEN GESTRICHEN.
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(LOESCH)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &KE &KL &NUMX
SET &KAMEL=&STR()
SET &N3=&STR(DATA)
WRITE DATEI :
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
IF &N1 = &STR(#) THEN GOTO FIN
WRITE KURVENZAHL =: ( < 51 )
WRITE > 0 : KURVEN EINZELN EINGEBEN ,
WRITE = 0 : KURVEN VOM ERSTEN BIS LETZTEN
READ &ANS
IF &STR(&ANS)=-&STR() THEN SET &INL=&ANS
IF &INL <= 0 THEN DO

```

91/02/22

```

GR: WRITE ERSTE UND LETZTE KURVE / &KE , &KL / =:
WRITE ( "ENTER"= JA | E = ERSTE | L = LETZTE KURVE )
READ &ANS
IF &ANS=&STR() THEN GOTO BEG
IF &ANS=&STR(E) THEN DO
  WRITENR ERSTE KURVE / &KE / =:
  READ &ANS
  IF &ANS~=&STR() THEN SET &KE=&ANS
  QUIT
IF &ANS=&STR(L) THEN DO
  WRITENR LETZTE KURVE / &KL / =:
  READ &ANS
  IF &ANS~=&STR() THEN SET &KL=&ANS
  QUIT
GOTO GR
QUIT
BEG: WRITE LETZTE KURVE, DIE BLEBEN SOLL: NUMMAX / &NUMX / =:
READ &ANS
IF &ANS~=&STR() THEN SET &NUMX=&ANS
SET &KAMEL=&KAMEL &KE &KL &NUMX &DR
EX '&SYSUID..CLIST(BEKI)' 'MEM(LOESCH)'
WRITE EINE KOPIE VON &TON
WRITE IM MASSENSPEICHER ABLEGEN ? ( "ENTER"= JA ) =:
READ &ANS
IF &ANS=&STR() THEN DO
  IF &SUBSTR(1:1,&TON)=&STR(I) THEN BS '&TON'
  IF &SUBSTR(1:1,&TON)=&STR(T) THEN TS '&TON'
  QUIT
IF &INL > 0 THEN SET &MAXL=&INL
IF &INL <=0 THEN SET &MAXL=&KL-&KE+1
IF &INL <=0 THEN SET &NU=&KE-1
SET &L=0
SET &K=0
DO WHILE &K < 5
  SET &K=&K+1
  SET &KB=&K*10
  SET &LU=&STR()
  DO WHILE &L<&KB
    IF &L >= &MAXL THEN GOTO EDI
    SET &L=&L+1
    IF &INL > 0 THEN DO
      WRITENR &L-TE KURVE =: ( "ENTER"= ENDE )
      READ &NU
      IF &NU = &STR() THEN GOTO EDI
      QUIT
    ELSE SET &NU=&NU+1
    SET &LU=&LU&STR(,)&NU
    SET &LAUF&K=&LU
    QUIT
  QUIT
EDI: IF &NU = &STR() THEN SET &L=&L-1
WRITE BEARBEITUNG DES AUFTRAGES :
EX '&SYSUID..CLIST(FEJ)'
EDIT &N3..CNTL OLD NUM
0120 //ERASE EXEC F7CLG,USER='INR105.SERVUS.LOAD'
0130 //C.SYSRINT DD DUMMY
0140 //C.SYSIN DD *
0150     PROGRAM LOESCH
0160     CALL ERASE

```

```

0170      STOP
0180      END
0190     /*
0191     //L.SYSPRINT DD DUMMY
0200     //L.SYSIN DD *
0210     INCLUDE SYSLIB(SERDIO,COPYDA,FENST4,NEXT)
0220     ENTRY LOESCH
0230     /*
0300     //G.FT11F001 DD DSN=&TON,DISP=SHR
0310     //G.FT15F001 DD UNIT=SYSDA,DCB=DCB.VBS,SPACE=(CYL,(20,10))
0320     //G.SYSPRINT DD SYSOUT=*
0330     //G.SYSIN DD *
0400     &&LALOES IRE=11,IWR=15,KPRI=1,FAKTOR=0.0,
0401     NIX=&LAUF1,
IF &L > 10 THEN 0402 &LAUF2,
IF &L > 20 THEN 0403 &LAUF3,
IF &L > 30 THEN 0404 &LAUF4,
IF &L > 40 THEN 0405 &LAUF5,
0409     NUMMAX=&NUMX, &&END
1999     &&LALOES IRE=0, &&END
2000     //
LIST 300 2000
EX '&SYSUID..CLIST(SUSA)' '&N3.'
FIN: EXIT

```

TEXIMA

TEXIMA wandelt Texte in GS-Bilder um. Die abzubildende Texte werden der Prozedur durch die MEMORY-Glieder tttt, eeee, ... zugeführt. TEXIMA kann bis zu 4 Texte in ein Bild einbringen. Die Bilder können DIN-Hoch oder -Querformat haben.

Die Prozedur arbeitet mit Hilfe der TRACEGS-Prozedur TEXT (s. M. Politzky, TRACEGS, Ein Programmsystem für die Zeichnungserstellung..., KFK 3237, 1981). Daraus folgt :

- die Texte müssen eine feste Satzlänge von 80 Zeichen haben,
- die jeweilige erste Zeile enthält 5 Steuergrößen - 3 beliebige REAL*4 Zahlen und zwei beliebige INTEGERS,
- die Zeichen "\$", "¢" und "#" werden als Steuerzeichen benutzt, so daß - falls man eine dieser Zeichen abbilden will - sie doppelt eigegeben muß, z.B. "\$\$" wird als "\$" gezeichnet.

Die Dateien werden verzerrungsfrei abgebildet, d.h. mehrere, hintereinander folgende Lehrstellen werden nicht zusammengelegt. Abweichend von der Eingabe&bsg. im TRACEGS-Handbuch dürfen Steuergrößen in jeder Text-Datei nur einmal - am Anfang - vorkommen.

Benötigte Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)
- inr000.CLIST(SERVIN)
- inr000.CLIST(TUKOR)

Benötigte Hilfsdateien :

- inr000.MEMORY(TEXIMA)
- inr000.memory(latexi)
- inr000.MEMORY(tttt)

- inr000.MEMORY(eeee)
- inr000.MEMORY(xxxx)
- inr000.MEMORY(tttt)

Die Prozedurliste:

```

PROC 0
CONTROL PROMPT MAIN MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON AP
WRITE
WRITE PROZEDUR      TEXIMA
WRITE TEXTDATEIEN WERDEN ZUR BILDER VERARBEITET.
WRITE
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(TEXIMA)'
SET &SYSDVAL=&KAMEL
READDVAL &AP &N1 &N2
EX '&SYSUID..CLIST(TUKOR)'
SET &KAMEL=&AP
WRITE "BUCH" ZUM BILDSPEICHERN :
SET &N3 = &STR(#)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1) CQ(1)'
IF &N1 = &STR(#) THEN GOTO FIN
SET &BUCH=&TON
EX '&SYSUID..CLIST(BEKI)' 'MEM(TEXIMA)'
EX '&SYSUID..CLIST(SERVIN)' 'TEXIMA'
IF &N1 = &STR(#) THEN GOTO FIN
IF &N1 < 1 THEN DO
    WRITE KEIN TEXTDATEI ALLOKIERT !
    GOTO FIN
    END
WRITE BEGINN DER PROZEDUR
IF &BUCH = &STR() THEN GOTO PI
IF &SYSDSN('&BUCH')=>OK THEN DO
    EX '&SYSUID..CLIST(ADA)' '&BUCH LIN(0) KA(10)'
    WRITE BUCH '&BUCH' ALLOKIERT.
    END
ALLOC DA('&BUCH') F(TRACEGS7) SHR REU
PI: SET &SYSDVAL = &STR(&AP &KAMEL &N1)
CALL 'INR105.SERVUS.LOAD(TEXIMA)' '&SYSDVAL '
FIN: WRITE PROZEDUR TEXIMA BEENDET.
EXIT

```

91/02/22

Hilfsprozeduren

Die Hilfsprozeduren erledigen Teilaufgaben, die in vielen verschiedenen Prozeduren in derselben Weise auftreten. Die Hilfsprozeduren können ihrerseits auch Hilfsprozeduren aufrufen.

Die Verständigung zwischen Hilfsprozedur und Prozedur erfolgt sowohl durch die Variablen im Kopf der Hilfsprozedur, als auch durch die, in der GLOBAL-Zeile aufgeführten Variablen.

ADA

ADA legt beim Aufruf unterschiedlich geartete Dateien an.
Die Eingabegrößen im Prozedurkopf sind :

DSNAM ist die Name der Datei,
LIN steuert Format und Blocklänge der Datei,
KA ist die Anzahl der "directory-blocks" der (gegliederten) Datei,
KA(0) => serielle Datei
DISK ist die Name der Platte wohin die neue Datei hingelegt werden soll.

Die Prozedurliste:

```
PROC 1 DSNAM LIN(80) KA(0) DISK(INR)
/* ALLOCATE
CONTROL PROMPT NOFLUSH MSG
IF &LIN = &STR(80) THEN +
ATTRIB JOKER RECL(&LIN) BLKSIZE(3120) RECFM(F B)
IF &LIN = &STR(132) THEN +
ATTRIB JOKER RECL(&LIN) BLKSIZE(3036) RECFM(F B)
IF &LIN = &STR(400) THEN +
ATTRIB JOKER RECL(&LIN) BLKSIZE(3200) RECFM(F B)
IF &LIN = &STR(133) THEN +
ATTRIB JOKER RECL(&LIN) BLKSIZE(3059) RECFM(F B A)
IF &LIN = &STR(180) THEN +
ATTRIB JOKER RECL(&LIN) BLKSIZE(32760) RECFM(V S)
IF &LIN = &STR(255) THEN +
ATTRIB JOKER RECL(&LIN) BLKSIZE(3120) RECFM(V B)
IF &LIN = &STR(X) THEN +
ATTRIB JOKER RECL(32760) BLKSIZE(19069) RECFM(V B S)
IF &LIN = &STR(0) THEN +
ATTRIB JOKER RECL(&LIN) BLKSIZE(19069) RECFM(U)
ALLOC DA('&DSNAM') NEW UNIT(&DISK) SPACE(10,2) TRA DIR(&KA) USING(JOKER)
FREE ATTRLIST(JOKER)
EXIT
```

92/03/06

ALOGS

ALOGS allokiert die Dateien, die bei einer G.S.-Sitzung in der Regel benötigt werden.
Am Ende der Sitzung werden die Dateien wieder befreit.

ALOGS benötigt die Hilfsdatei

- inr000.MEMORY(BOOKS)

sowie die Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(NEVE)
- inr000.CLIST(TUKOR)

Die Prozedurliste:

```

PROC 0
/* ALLOCATE G.S. DATASETS                                91/03/12
CONTROL PROMPT NOFLUSH MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON AP
IF &EA=0 THEN GOTO K5
EX '&SYSUID..CLIST(BEKI)' 'MEM(BOOKS)'
SET &NOTNAGL=&KAMEL
SET &SYSDVAL=&KAMEL
READDVAL &AP &SIZE &PRO1 &BUCH1 &PROU &UFOS &SF &PRO2 &BUCH2
EX '&SYSUID..CLIST(TUKOR)'
SS: IF &DATATYPE(&SIZE) = NUM THEN DO
    WRITE SIZE MUSS EINE POSITIVE ZAHL SEIN !
    WRITE SIZE / &SIZE / =:
    READ &SIZE
    GOTO SS
    END
ALLOC FI(COMM) DS(*)
ALLOC FI(ADMDEFS) DA('TSOSYS.ADMDEFS') SHR REU
IF &AP=&STR(PCG) THEN +
ALLOC F(ADMPG) DA('TSOSYS.GDDM.GDDMSAM') SHR
IF &AP=&STR(GAT) THEN +
ALLOC FI(GAAPLIB) DS('TSOSYS.GAS.FONT') SHR REU
IF &AP=&STR(T15) THEN GSA
IF &AP=&STR(GAT) && &AP = &STR(T15) THEN +
ALLOC FI(ADMSYMBL) DS('TSOSYS.GDDM.GDDMSYM') SHR REU
IF &AP=&STR(T25) THEN +
ALLOC FI(GSLINK) DS('TS0186.TGS.LOAD') SHR REU
SET &KAMEL=&AP &SIZE
WRITE GS BILDER-DATEI :
SET &N1=&PRO1
SET &N2=&BUCH1
SET &N3=&STR(#)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
IF &N1=&STR(#) THEN SET &SIZE=0
IF &SIZE=0 THEN DO
    SET &KAMEL=&NOTNAGL
    GOTO K3
    END
SET &GSB=&TON
ALLOC FI(B1) DA('&GSB') SHR REU
WRITE GS UFO-DATEI :
SET &N1=&PROU
SET &N2=&UFOS
SET &N3=&STR(LOAD)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
SET &UFB=&TON
IF &UFB=&STR() THEN ALLOC FI(UFOLIB) DS('&UFB') SHR REU
ELSE SET &UFB=&STR(#)
WRITE ZUSATZDATEI / &SF / =: ( # = "KEINE" | G = "GDF"-DATEI )
READ &ANS
IF &ANS=&STR() THEN SET &SF=&ANS
IF &SF=&STR(#) && &SF=&STR(G) THEN SET &SF=&STR(B)

```

```

SET &KAMEL=&KAMEL &SF
IF &SF=&STR(#) THEN DO
    SET &KAMEL=&KAMEL &PRO2 &BUCH2
    GOTO K3
END
SET &N1=&PRO2
SET &N2=&BUCH2
SET &N3=&STR(#)
EX '&SYSUID..CLIST(NEVE)' 'LQ(1)'
IF &TON=&STR() THEN SET &SF=&STR(#)
IF &SF=&STR(B) THEN ALLOC FI(B2) DA('&TON') SHR REU
IF &SF=&STR(G) THEN ALLOC FI(ADMGDF) DA('&TON') SHR REU
K3: SET &GOBLIN=&SIZE &GSB &UFB &SF &TON
EX '&SYSUID..CLIST(BEKI)' 'MEM(BOOKS)'
SET &KAMEL=&GOBLIN
GOTO FIN
K5: FREE FI(B1,COMM,ADMDEFS)
SET &SYSDVAL=&KAMEL
READDVAL &SIZE &GSB &UFB &SF &ZUB
IF &AP=&STR(PCG) THEN FREE F(ADMPC)
IF &AP=&STR(GAT) THEN FREE FI(GAAPLIB)
IF &AP=&STR(GAT) && &AP=&STR(T15) THEN FREE FI(ADMSYMBL)
IF &AP=&STR(T25) THEN FREE FI(GSLINK)
IF &UFB=&STR(#) THEN FREE FI(UFOLIB)
IF &SF=&STR(#) THEN DO
    IF &SF=&STR(B) THEN FREE FI(B2)
    IF &SF=&STR(G) THEN FREE FI(ADMGDF)
END
FIN: EXIT

```

BEKI

BEKI liest - falls EA = 0 ist - Eingangswerte einer Anzahl von Prozedurgrößen von der Datei inr000.MEMORY(MEM) und speichert - falls EA = 1 ist - dieselbe Größen als Ausgangswerte in die Datei. Falls inr000.MEMORY(MEM) nicht vorhanden, legt BEKI sie an. Dazu benötigt BEKI die Hilfsprozedur inr000.CLIST(ADA) .

Die Prozedurliste:

```

PROC 0 MEM()
/* INPUT-OUTPUT
CONTROL PROMPT NOFLUSH MSG END(QUIT)
GLOBAL KAMEL EA
SET &DAM = &STR(&SYSUID..MEMORY)
IF &SYSDSN('&DAM.(&MEM)') = OK THEN GOTO AL
IF &SYSDSN('&DAM') = OK THEN GOTO CM
EX '&SYSUID..CLIST(ADA)' '&DAM LIN(80) KA(10)'
WRITE '&DAM' ALLOCATED
CM: IF &SYSDSN('&DAM.(&MEM)') = OK THEN DO
    WRITE REQUIRED MEMBER NOT FOUND, IT WILL BE CREATED .
    EDIT '&DAM.(&MEM) ' NEW FORTGI NUM
    10 ???
    LIST
    UNNUM
    END SAVE
    QUIT
AL: ALLOC FILE(MEMORY) DA('&DAM.(&MEM)') OLD
IF &EA=1 THEN GOTO IN

```

91/02/20

```

WRITE LESEN AUS '&DAM.(&MEM) '
ERROR DO
    WRITE MEMBER IS EMPTY
    GOTO CL
    QUIT
OPENFILE MEMORY
GETFILE MEMORY
SET &KAMEL=&STR(&MEMORY)
CL: CLOSFILE MEMORY
SET &EA=1
FREE DA('&DAM.(&MEM) ')
GOTO FIN
IN: SET &L = &LENGTH(&STR(&KAMEL))
OPENFILE MEMORY OUTPUT
SET &MEMORY=&SUBSTR(1:&L,&KAMEL)
PUTFILE MEMORY
CLOSFILE MEMORY
FREE DA('&DAM.(&MEM) ')
WRITE SCHREIBEN IN '&DAM.(&MEM) '
FIN: EXIT

```

DRUQQ

DRUQQ schreibt - mit Hilfe des Benutzers - Jobkarten für die IEBTPCH - Prozedur.
 DRUQQ benötigt weder eine Hilfsdatei noch eine Hilfsprozedur.
 Die Prozedurliste:

```

PROC 0
/* CREATES IEBTPCH - INPUT                                91/02/21
CONTROL PROMPT NOFLUSH MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE
WRITE EINGABEN FUER DIE IEBTPCH - PROZEDUR :
WRITE ZEILENLAENGE =: ( ENTER = 80 | SONST = 120 )
READ &ANS
IF &ANS = &STR() THEN SET &N2 = &STR(80)
IF &ANS ^= &STR() THEN SET &N2 = &STR(120)
WRITE VORSCHUBZEICHEN =: ( ENTER = KEINE | SONST = ASA )
READ &ANS
IF &ANS = &STR() THEN SET &M1=&STR()
ELSE SET &M1=&STR(,PREFORM=A)
WRITE HERSTELLUNG DER CNTL-DATEI:
EDIT '&SYSUID..&N3..CNTL' OLD CNTL NUM
210 //PRINT EXEC PGM=IEBTPCH
220 //SYSPRINT DD DUMMY
230 //SYSUT1 DD DISP=SHR,DSN=&TON
250 //SYSUT2 DD SYSOUT=&N1
260 //SYSIN DD *
END SAVE
EXIT

```

FEJ

FEJ schreibt Jobkarten für CNTL-Prozeduren. Der Benutzer möge den Prozedurkopf mit den eigenen Angaben ausfüllen. Auch in der Karte 0111 - am Ende der Prozedur - soll man anstatt Pinr1 die zutreffende Nummer des Ausgabegerätes eintragen.

Mit der Eingabegröße **R** kann man den REGION-Parameter ändern, mit **NAME** kann man einen Namen für die CNTL-Datei wählen.

FEJ benötigt die Hilfsprozeduren inr000.CLIST(ADA) und inr000.CLIST(BEKI).

Die Prozedurliste:

```
PROC 0 R(1000) KOR(NEW) NAME() +
USER(dombrowsky) ACCOUNT('0000,999,abcde')
/* PREPARES JOBCARDS
CONTROL PROMPT NOFLUSH MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 N3 M1 TON
WRITE ZUSAMMENSTELLUNG DER JOB-KARTE(-N)
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(FEJ)'
SET &SYSDVAL=&KAMEL
READDVAL &N1 &N2 &N3 &US &AC &MA &LI
IF &NAME = &STR() THEN SET &N3=&NAME
SET &DAC = &SYSUID..&N3..CNTL
NY: WRITE DATEINAME ( "ENTER" = ES BLEIBT DIE ALTE DATEI )
READ &ANS
IF &ANS = &STR() THEN GOTO ED
IF &ANS= &STR() THEN SET &N3=&ANS
WRITE DRUCKQUALITAET / &N1 / =: ( H | A | W = TSO-ZEICHENSATZ )
READ &ANS
IF &ANS = &STR() THEN SET &N1= &ANS
WRITENR JOB-KENNZEICHNUNG / &N2 / =:
READ &ANS
IF &ANS = &STR() THEN SET &N2=&ANS
IF &LENGTH(&N2) > 2 THEN SET &N2 = &SUBSTR(1:2,&N2)
IF &US = &STR() THEN SET &USER=&US
WRITENR "USER" / &USER / =: ( "ENTER"=JA )
READ &ANS
IF &ANS = &STR() THEN SET &USER=&ANS
IF &AC = &STR() THEN SET &ACCOUNT=&AC
WRITE "ACCOUNT" / &ACCOUNT / =: ( "ENTER"=JA )
WRITE NEUE ZEICHENKETTE MIT HOCHKOMMATA ('0123,456,ABCDE') EINGEBEN !
READ &ANS
IF &ANS = &STR() THEN SET &ACCOUNT=&ANS
WRITENR "LINES" ? =: ( "ENTER"=KEINE | 10 | ... )
READ &LI
SET &MA=&LI
IF &LI = &STR() THEN DO
    SET &MA=&STR(L)
    GOTO DAI
QUIT
WRITE DRUCKORT / &00 / =: ( "ENTER"=NORM | 0 = PINR1 )
READ &ANS
IF &ANS = &STR(0) THEN SET &MA=&STR()
DAI: IF &N3 = &STR() THEN SET &N3=&STR(CLIST)
OK: SET &DAC = &SYSUID..&N3..CNTL
    IF &SYSDSN('&DAC.') = OK THEN DO
        WRITE DATEI '&DAC' BEREITS VORHANDEN !
        WRITE SOLLTE SIE UEBERSCHRIEBEN WERDEN ? =: +
            ( "ENTER"=JA | NEUE NAME )
```

```

READ &ANS
IF &ANS ^= &STR() THEN DO
    SET &N3 = &ANS
    GOTO OK
    QUIT
ELSE DO
    EDIT '&DAC.' OLD NUM
    RENUM
    DEL 10 9990
    END SAVE
    SET &KOR=&STR(OLD)
    QUIT
    QUIT
ED: IF &SYSDSN('&DAC') ^= OK THEN SET &KOR = &STR(NEW)
SET &KAMEL = &STR(&N1 &N2 &N3 &USER &STR('&ACCOUNT') &MA &LI)
EX '&SYSUID..CLIST(BEKI)' 'MEM(FEJ)'
IF &KOR = &STR(NEW) THEN ERASE '&DAC'
EDIT '&DAC.' &KOR CNTL NUM
0100 //&SYSUID.&N2 JOB (&ACCOUNT),&USER,MSGLEVEL=(1,1),REGION=&R.K,
0110 // NOTIFY=&SYSUID,MSGCLASS=&N1
IF &MA = &STR(L) THEN 0111 &STR(//&STR(*MAIN LINES=&LI))
IF &MA = &STR(O) THEN 0111 /*FORMAT PR,DDNAME=,DEST=Pinr1
END SAVE
EXIT

```

GLIEDER

Diese Hilfsprozedur listet die Glieder einer Datei "DSNAM" am Bildschirm. DSNAM muß der vollständige Name sein.

Die Prozedurliste:

```

PROC 1 DSNAM
/* MEMBERS
CONTROL PROMPT NOFLUSH MSG
IF &SYSDSN('&DSNAM') ^= OK THEN DO
    WRITE DATEI '&DSNAM' GIBT ES NICHT. !
    EXIT
    END
WRITE GLIEDER DER DATEI '&DSNAM'
SET &SYSOUTTRAP = 200
LISTDS '&DSNAM' MEMBERS
SET I = 7
SET M = 0
DO WHILE &I <= &SYSOUTLINE
    SET &M = &M+1
    SET DSN = &&SYSOUTLINE&I
    WRITE &M &DSN
    SET &I = &I+1
    END
EXIT

```

KRAM

KRAM komprimiert die fast täglich benutzte Dateien MEMORY, NAPLO.DATA und CLIST. Mit der eingabegröße **IND** kann man die Prozedur steuern : bei einer "Y" als erste Zeichen des Wortes IND wird die erste Datei bearbeitet usw. Bei der Wahl IND(NNY) wird z.B. nur die CLIST-Datei bearbeitet.

Weder Hilfsprozeduren noch Hilfsdateien werden benötigt.

Die Prozedurliste:

```
PROC 0 IND(NNY)
/* COMPRESS/REDUCE SLAVING DATASETS                               91/09/20
CONTROL PROMPT MAIN MSG END(QUIT)
SET I=0
SET &L=&LENGTH(&IND)
BL: DO WHILE &I < &L
  SET &I=&I+1
  SET &X=&SUBSTR(&I:&I,&IND)
  IF &I = 1 THEN SET &TON=&STR(&SYSUID..MEMORY)
  IF &I = 2 THEN SET &TON=&STR(&SYSUID..NAPLO.DATA)
  IF &I = 3 THEN SET &TON=&STR(&SYSUID..CLIST)
  IF &X = &STR(Y) THEN GOTO BL
  TR: IF &SYSDSN('&TON') = OK THEN DO
    COMPRESS '&TON' NOLIST
    REDUCE '&TON'
    WRITE '&TON' HAS BEEN TREATED
  QUIT
  GOTO BL
QUIT
EXIT
```

NEVE

NEVE hilft dem Benutzer beim Zusammenstellen des Namens einer Datei und prüft, ob die genannte Datei vorhanden ist oder nicht.

Zu den Eingabegrößen LQ, CQ, SQ :

LQ wählt die Namensteile, die erfragt werden sollen, bei
LQ = 1 werden "Projekt" und "Name" erfragt, bei
LQ = 2 wird nur "Name" erfragt und bei
LQ = 3 werden nur "Name" und "Typ" ermittelt. Bei
LQ = 0 werden alle drei Namensteile erfragt.

CQ steuert die Datei-Kontrolle: bei
CQ = 0 wird überprüft, ob die Datei vorhanden sei, bei
CQ ≠ 0 erfolgt keine Kontrolle.

SQ steuert die Registratur : bei
SQ = 0 werden die erfragten Namensteilen in die "MEMORY" eingetragen, bei
SQ ≠ 0 dagegen nicht.

Die Prozedurliste:

```

PROC 0 LQ(0) CQ(0) SQ(0)
CONTROL PROMPT NOFLUSH MSG
/* NAME OF A DATASET
91/03/08
GLOBAL KAMEL EA N1 N2 N3 M1 TON
SET &NOTA=&STR(MEMBER SPECIFIED, BUT DATASET IS NOT PARTITIONED)
IF &N1 = &STR() | &N1 = &STR(#) THEN SET &N1=&SYSUID
IF &M1 = &STR() | &M1 = &STR(#) | +
&DATATYPE(&M1)=&STR(CHAR) THEN SET &M1=&STR(TEMP)
PRJ: SET &TON = &STR()
IF &LQ > 1 THEN GOTO NAM
WRITENR PROJECT / &N1 / =: ( "ENTER"=JA | # = KEINE DATEI )
READ &A1
IF &A1 = &STR() THEN DO
  IF &A1 = &STR(#) THEN GOTO REG
  SET &N1 = &A1
  IF &N1 = &STR(I) THEN SET &N1 = &STR(&SYSUID)
  END
NAM: WRITENR NAME / &N2 / =: ( "ENTER" (=JA) )
READ &AN
IF &AN = &STR() THEN SET &N2 = &AN
TYP: IF &LQ = 0 | &LQ = 3 THEN DO
  WRITENR TYP / &N3 / =: ( "ENTER" (=JA) | # (=TYP FEHLT) )
  READ &AN
  IF &AN = &STR() THEN SET &N3 = &AN
  END
IF &N3 = &STR(#) THEN SET &TON= &N1..&N2..&N3
  ELSE SET &TON= &N1..&N2
CTR: IF &CQ = 0 THEN GOTO REG
IF &SYSDSN('&TON')=OK THEN DO
  WRITE &TON &SYSDSN('&TON') !
  WRITE
  WRITE KATALOG DER &N1 -DATEIEN :
  WRITE
  CAT &N1
  GOTO PRJ
  END
IF &SYSDSN('&TON(&M1)')=&NOTA THEN DO
  WRITE &SUBSTR(22:48,&NOTA)
  SET &M1=&STR(#)
  END
REG: IF &SQ = 0 THEN GOTO FF
IF &LQ < 2 THEN SET &KAMEL=&KAMEL &N1
SET &KAMEL=&KAMEL &N2
IF &LQ = 0 | &LQ = 3 THEN SET &KAMEL=&KAMEL &N3
FF: IF &A1 = &STR(#) THEN SET &N1=&A1
EXIT

```

OBJEKT

OBJEKT legt einen Teil eines Bildes für die weitere Bearbeitung fest. Die Hilfsprozedur unterstützt die Wahl des Fensters, zweier Arbeitsfarben sowie die Vergrößerung eines Teilbildes.

Die Prozedurliste:


```

PROC 0 X1U(0.0) Y1U(0.0) X20(11.7) Y20(8.28)
/* SELECT PART OF A PICTURE
CONTROL PROMPT NOFLUSH MSG END(QUIT)
GLOBAL KAMEL EA N1 N2 PAN BIN TON GER
SET &SYSDVAL=&KAMEL
READDVAL &FORM &MG &MH &XLU &YLU &XRO &YRO
IF &XLU=&STR() THEN SET &XLU=&X1U
IF &YLU=&STR() THEN SET &YLU=&Y1U
IF &XRO=&STR() THEN SET &XRO=&X20
IF &YRO=&STR() THEN SET &YRO=&Y20
NBO: WRITE OBJEKT " &BIN..&PAN " IN ORDNUNG ?
WRITE "ENTER" = JA
WRITE "B" = NEUES BILD
WRITE "T" = NEUES TEILBILD
WRITE "C" = ARBEITSFARBEN WAHLEN
WRITE "W" = FENSTER
READ &ANS
IF &ANS=&STR() THEN GOTO FIN
IF &ANS=-&STR() THEN SET &N1=&ANS
IF &N1=&STR(B) THEN DO
    SET &LAB=&BIN
    EX '&SYSUID..CLIST(TAGOK)'
    IF &BIN=&STR(!) THEN DO
        SET &N1=&BIN
        SET &BIN=&LAB
    QUIT
QUIT
IF &N1=&STR(!) THEN EXIT
IF &ANS=&STR(W) THEN GOTO KF
IF &ANS=&STR(T) THEN GOTO TBI
IF &ANS=&STR(C) THEN DO
    NCO: WRITENR GRUNDFARBE / &MG / =: ( 0 , ... , 7 )
    READ &ANS
    IF &ANS > 7 THEN GOTO NCO
    IF &ANS -=&STR() THEN SET &MG=&ANS
    WRITE GRUNDFARBE = C&MG
    NC1: WRITENR FARBE ZUM HERVORHEBEN / &MH / =: ( 0 , ... , 7 )
    READ &ANS
    IF &ANS > 7 THEN GOTO NC1
    IF &ANS -=&STR() THEN SET &MH=&ANS
    WRITE HERVORHEBEN MIT DER FARBE C&MH
    QUIT
GOTO NBO
IF &ANS=&STR(Z) THEN DO
    KF: WRITE FENSTER IM AUGENBLICK :
    WRITE (&XLU,&YLU) - (&XRO,&YRO), DIN-&FORM FORMAT
    WRITE IN ORDNUNG =:
    WRITE "ENTER" = JA
    WRITE "L" = LINKSUNTEN AENDERN
    WRITE "R" = RECHTSOBEN AENDERN
    WRITE "F" = FENSTERFORM AENDERN
    WRITE "SONST" = NATUERLICHES FENSTER
    READ &ANS
    IF &ANS=&STR() THEN GOTO FIN
    IF &ANS-=&STR(L) && &ANS-=&STR(R) && &ANS-=&STR(F) THEN DO
        SET &XLU = &STR()
        SET &YLU = &STR()
        SET &XRO = &STR()
        SET &YRO = &STR()

```

```

        GOTO FIN
        QUIT
    IF &ANS=&STR(F) THEN DO
        IK: WRITENR ( "H" = DIN-H | "Q" = DIN-Q )
        READ &ANS
        IF &ANS=&STR() THEN SET &FORM=&ANS
        IF &FORM=&STR(H) && &FORM=&STR(Q) THEN GOTO IK
        QUIT
    IF &ANS=&STR(L) THEN DO
        WRITENR ECKE LINKSUNTEN : X / &XLU / =: , Y / &YLU / =:
        READ &AX &AY
        IF &AX=&STR() THEN SET &XLU = &STR(&AX)
        IF &AY=&STR() THEN SET &YLU = &STR(&AY)
        QUIT
    IF &ANS=&STR(R) THEN DO
        WRITENR ECKE RECHTSOBEN : X / &XRO / =: , Y / &YRO / =:
        READ &AX &AY
        IF &AX=&STR() THEN SET &XRO = &STR(&AX)
        IF &AY=&STR() THEN SET &YRO = &STR(&AY)
        QUIT
    GOTO KF
    QUIT
TBI: WRITE NAME DES TEILBILDES / &PAN / =:
WRITE      "ENTER" : JA
WRITE      ? : BILD-STRUKTUR
WRITE      # : KEIN TEILBILD
READ &ANS
IF &ANS = &STR(?) THEN DO
    INITGS 500
    OW &GER
    RPIB &BIN B1
    LS
    ENDGS
    GOTO TBI
    QUIT
IF &ANS=&STR() THEN SET &PAN=&ANS
GOTO NBO
FIN: SET &KAMEL=&FORM &MG &MH &XLU &YLU &XRO &YRO
EXIT

```

SERVIN

SERVIN bereitet die Herstellung eines SERVUS-Bildes vor.

Zuerst erfragt die Prozedur vom Benutzer verschiedene bildspezifische Eingabedaten (DIN-Format, KfK-Emblem, ...) sowie die Namen der für die Bildherstellung benötigten Hilfsdateien. Dann erfolgt das Anknüpfen dieser Dateien und die vorgesehene FORT-RAN-Prozedur wird aufgerufen.

LANAM ist die Name der FORTRAN-Prozedur.

SERVIN benötigt die Hilfsdatei inr000.MEMORY(SERVIN) und folgende Hilfsprozeduren :

- inr000.CLIST(ADA)
- inr000.CLIST(BEKI)
- inr000.CLIST(GLIEDER)
- inr000.CLIST(NEVE)
- inr000.CLIST(TAGOK)

Die Prozedurliste:

```

PROC 1 LANAM
/* PREPARES DATA FOR A SERVUS-PICTURE CONTROL PROMPT MSG
GLOBAL KAMEL EA N1 N2 N3 M1 TON
SET &DRA=&STR(LA&SUBSTR(1:4,&LANAM))
SET &EA=0
EX '&SYSUID..CLIST(BEKI)' 'MEM(SERVIN)'
SET &SYSDVAL=&KAMEL
READDVAL &IDIN &FEDA &BILN &ABT &CALE +
&N1 &N2 &N3 &M1 &Q1 &Q2 &Q3 &Q4
WRITE FENSTER / &IDIN / =: ( 1 : DIN-HOCH | SONST : DIN QUER )
READ &ANS
IF &ANS = &STR() THEN SET &IDIN=&ANS
IF &IDIN = &STR(1) THEN SET &IDIN = &STR(0)
WRITE QUELLE DER RAHMENDATEN / &FEDA / =:
WRITE      0 : DATEI &DRA
WRITE      SONST : &LANAM-VORGABE
READ &ANS
IF &ANS = &STR() THEN SET &FEDA=&ANS
IF &FEDA = &STR(0) THEN SET &FEDA=&STR(1)
BIN: WRITE VORGESEHENE BILDNAME / &BILN / =:
READ &ANS
IF &ANS = &STR() THEN SET &BILN=&ANS
SET &L=&LENGTH(&BILN)
IF &L > 8 THEN SET &BILN=&SUBSTR(1:8,BILN)
IF &L < 6 THEN DO
    WRITE ZU KURZ ! BITTE MINDESTENS 6 ZEICHEN .
    GOTO BIN
END
WRITE BILDER MIT KFK-SYMBOL / &ABT / =: +
( "?" : NEIN | INSTITUTSKUERZEL )
READ &ANS
IF &ANS = &STR() THEN DO
    SET &L=&LENGTH(&ANS)
    IF &L > 4 THEN SET &L = 4
    SET &ABT=&SUBSTR(1:&L,&ANS)
END
WRITE BILDER MIT DATUM / &CALE / =: ( "0" : JA | SONST NEIN )
READ &ANS
IF &ANS = &STR() THEN SET &CALE=&ANS
IF &CALE = &STR(0) THEN SET &CALE=&STR(9)
SET &TEVE=&STR(&IDIN &FEDA &BILN &ABT &CALE)
SET &KAMEL=&TEVE
WRITE ANGABEN ZUR QUELLDATEI &DRA :
EX '&SYSUID..CLIST(NEVE)'
IF &N1 = &STR(#) THEN GOTO FIN
IF &M1 = &STR(#) THEN DO
    EX '&SYSUID..CLIST(TAGOK)'
    IF &M1 = &STR(!) THEN GOTO FIN
    SET &TON=&STR(&TON(&M1))
END
ALLOC DA('&TON') F(FT10F001) SHR REU
SET &KAMEL = &STR(&KAMEL &M1)
WRITE ANGABEN ZUR ZUSAETZLICHEN TEXT-DATEIEN ABB1 ... ABB4 :
SET &TON=&SYSUID..MEMORY
SET &I=0
SET &N=0

```

91/03/08

```

DO WHILE &I < 4
  SET &I=&I+1
  SET MM = &&Q&I
  IF &MM ^= &STR() THEN SET &M1 = &MM
  ELSE SET &M1 = &STR(TEMPNAME)
  EX '&SYSUID..CLIST(TAGOK)'
  IF &M1 = &STR(!) THEN GOTO FIE
  SET &FI = &STR(FT3)&STR(&I)&STR(F001)
  ALLOC DA('&TON(&M1)') F(&FI) SHR REU
  SET &KAMEL = &KAMEL &M1
  SET &N=&I
  END
FIE: EX '&SYSUID..CLIST(BEKI)' 'MEM(SERVIN)'
SET &N1=&N
SET &KAMEL=&TEVE
ALLOC DA(*) F(COMM) SHR REU
ALLOC FI(ADMDEFS) DA('TSOSYS.ADMDEFS') SHR REU
FIN: EXIT

```

TAGOK

TAGOK hilft dem Benutzer beim aufsuchen eines Gliedes einer gegliederten Datei. Falls CQ = 0 ist, dann wird auch noch überprüft, ob das Glied vorhanden sei. TAGOK selber benötigt eine Hilfsprozedur : inr000.CLIST(GLIEDER) .

Die Prozedurliste:

```

PROC 0 CQ(0)
CONTROL PROMPT NOFLUSH MSG
/* MEMBERS
GLOBAL KAMEL EA N1 N2 N3 M1 TON
ANF: WRITE GLIED / &M1 / +
( "ENTER" = OK | ? = "LISTE" | ! = "ENDE" ) =:
READ &ANS
IF &ANS=&STR(?) THEN DO
  EX '&SYSUID..CLIST(GLIEDER)' '&TON'
  GOTO ANF
  END
IF &ANS^=&STR() THEN SET &M1 = &ANS
IF &ANS=&STR(!) THEN EXIT
IF &SYSDSN('&TON.(&M1)')^-OK && &CQ = 0 THEN DO
  WRITE NICHT VORHANDEN !
  GOTO ANF
  END
EXIT

```

91/02/20

TUKOR

TUKOR unterstützt den Benutzer bei der Wahl eines graphischen Arbeitsplatzes.

Die Prozedurliste:

PROC 0

91/02/20

/* GRABS A TERMINAL

CONTROL PROMPT NOFLUSH MSG

GLOBAL KAMEL EA N1 N2 N3 M1 TON AP

SET &N3 = 0

IF &AP = 2 THEN SET &AP=&STR(327)

IF &AP = 3 THEN SET &AP=&STR(GAT)

IF &AP = 4 THEN SET &AP=&STR(TCX)

IF &AP = 5 THEN SET &AP=&STR(317)

IG: WRITE

WRITE GRAPHISCHER ARBEITSPLATZ / &AP / =:

WRITE "PCG" = IBM 5378 PC

WRITE "327" = IBM 3279 G | MEMOREX 2079

WRITE "GAT" = IBM 3277 GA

WRITE "T15" = TEKTRONIX 4015

WRITE "TCX" = TEKTRONIX CX 4109 A

WRITE "T25" = TEKTRONIX 4125

WRITE "317" = IBM 3179 G

READ &ANS

IF &ANS = &STR() THEN SET &AP=&ANS

IF &AP = &STR(GAT) THEN SET &N3=3

IF &AP = &STR(PCG) | &AP = &STR(317) THEN SET &N3=5

IF &AP = &STR(327) THEN SET &N3=2

IF &AP = &STR(T15) | &AP = &STR(TCX) | +

&AP = &STR(T25) THEN SET &N3=4

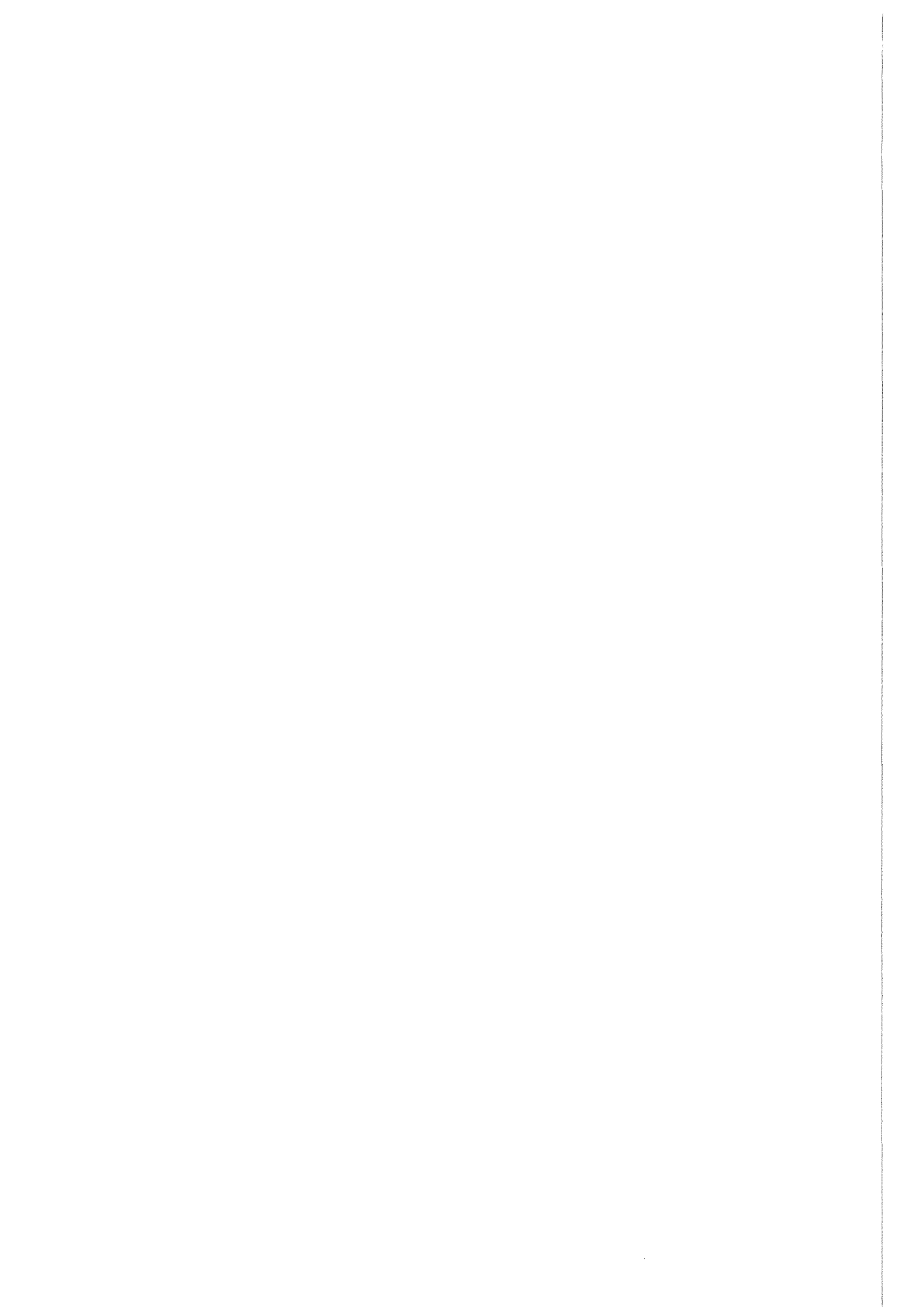
IF &N3 < 2 | &N3 > 5 THEN DO

WRITE GIBT'S NICHT !

GOTO IG:

END

EXIT



Diese Prozeduren

- stellen die CNTL-Prozeduren fertig, oder
- beseitigen nicht mehr benötigten Dateien.

HILDA

HILDA kümmert sich um die Hilfsdateien der GDDM-Prozedur : speichert sie, holt sie aus dem Massenspeicher zurück oder löscht sie.

Die Prozedurliste:

```
PROC 0
CONTROL PROMPT MAIN MSG
WRITE DIE HILFS-DATEIEN DER GDDM-PROZEDUR          91/03/08
WRITE SPEICHERN, ZURUECKHOLEN ODER LOESCHEN ( S | B | # ) =:
READ &ANS
IF &ANS = &STR(#) && &ANS = &STR(B) && &ANS = &STR(S) +
THEN SET &ANS = &STR(S)
IF &ANS = &STR(B) | &ANS = &STR(S) THEN DO
    B&ANS ADMCDATA WAIT
    B&ANS ADMCFORM WAIT
    IF &ANS = &STR(B) THEN DO
        FREE ATTR(DCB)
        ATTRIB DCB RECFM(F B) LRECL(400) BLKSIZE(3200)
        ALLOC DA(ADMSYMBL) NEW DIR(10) SP(1,1) TRACKS USING(DCB)
        FREE ATTR(DCB)
        REDUCE ADMCDATA
        REDUCE ADMCFORM
    END
END
ELSE DO
    ERASE ADMCDATA
    ERASE ADMCFORM
    ERASE ADMSYMBL
    ERASE ADMCDEF
END
EXIT
```

SUSA

SUSA speichert die fertiggestellte CNTL-Prozedur und auf Verlangen lässt sie den Auftrag ausführen.

Die Prozedurliste:

```
PROC 1 NAME
CONTROL PROMPT NOFLUSH MSG END(QUIT)
/* FINISHES THE JOBCARDS.          91/02/20
GLOBAL KAMEL EA N1 N2 N3 M1 TON
UNNUM
SET &TON = &SYSUID..&N3..CNTL
SAVE '&TON'
```

```
WRITE GESPEICHERT ALS '&TON'  
WRITE JOB ABGEBEN ? ( "ENTER"=JA ) =:  
READ &ANS  
IF &ANS = &STR() THEN SUBMIT '&TON'  
END NOSAVE  
EXIT
```

TERI

Teri räumt die Dateien, die im Rahmen einer Textausgabe von GML angeschafft wurden auf.

Die Prozedurliste:

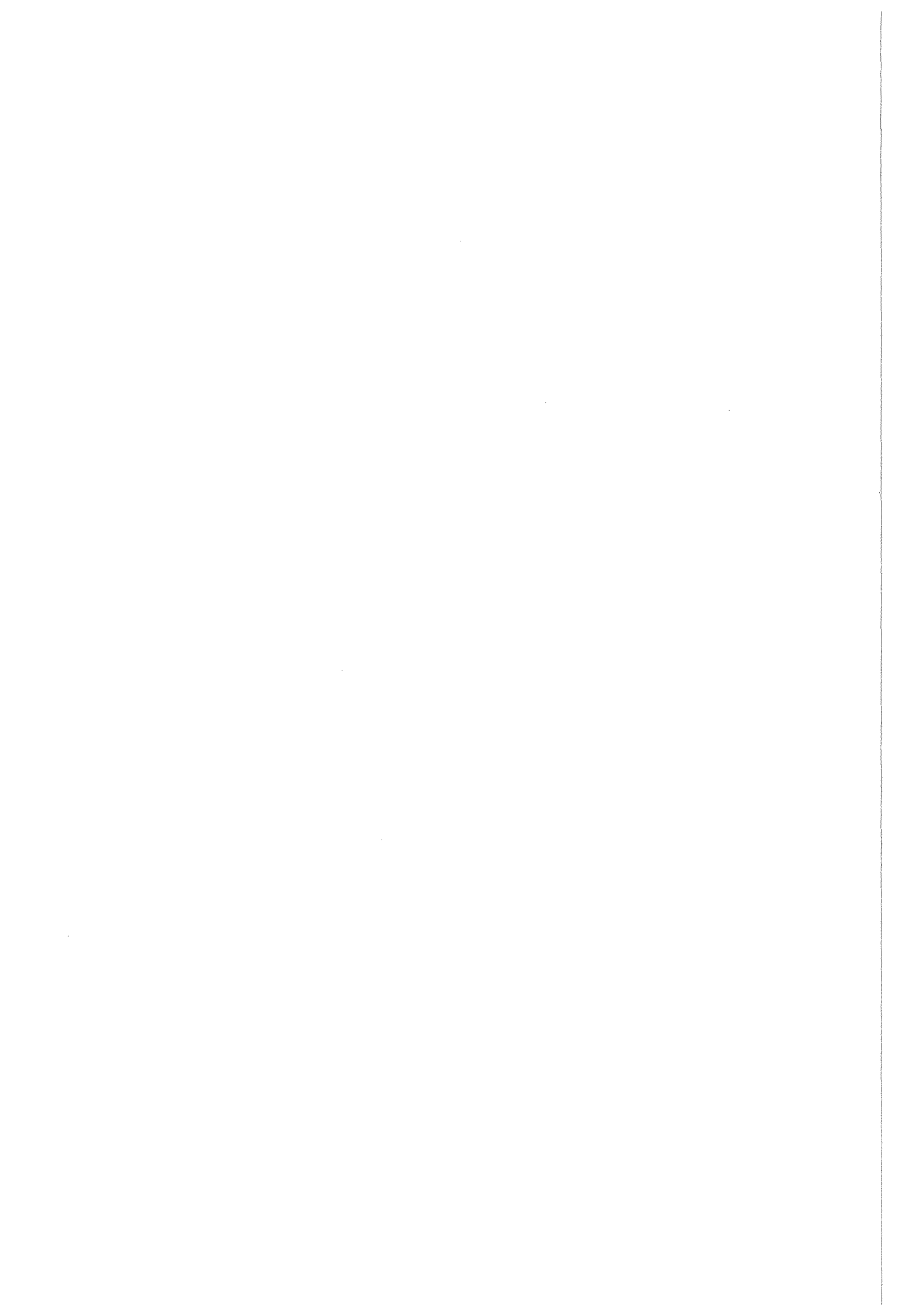
```
PROC 0  
GLOBAL KAMEL EA N1 N2 N3 M1 TON  
/* MOPS UP  
ERASE DSMUTTOC.TEXT  
ERASE GML.LIST  
ERASE EML.LIST  
ERASE EML.LISTING  
EXIT
```

91/02/20

Literatur:

- /1/ IBM TSO Extensions CLISTs. SC28-1304-3, 4th Ed., 1987
- /2/ D. B. Engle, Spool Display Facility, Version 2.9, Users Guide
Bank of America, Software Engineering Services, Concord CA, 1990.
GML-Datei zur Druckausgabe :
- 'TSOSYS.SDF.TEXT(SDFUSER)'
- /3/ Siemens System 7-800 FORTRAN77 User's Guide. Edition December 1985.
Chapt. 11.6 Execution of FORTRAN programs, P. 424
- /4/ S. Cohen, S. C. Pieper, The Speakeasy-3 Reference Manual,
Argonne National Laboratory, Dec. 1977.
- /5/ H.D.I., GS-Handbuch. Unveröffentlichter Bericht des KfK, Juli 1991.
- /6/ K. Thurnay, SERVUS. Ein System von Dateien und Prozeduren zur Speicherung
und Präsentation von Rechenergebnissen, die Funktionen (einer Variablen) sind.
Version V, April 1989 Kfk 4587, Juli 1989.
- GML-Datei zur Druckausgabe (Version V.1 , 1991) :

'INR105.SERVUS.TEXT(\$)'



Anhang A. Die Datei "Memory"

DATA SET NAME: INR105.MEMORY

GENERAL DATA:

Volume serial: INR004
 Device type: 3390
 Organization: P0
 Record format: FB
 Record length: 80
 Block size: 3120
 1st extent Tracks: 2
 Secondary Tracks: 10

CURRENT ALLOCATION:

Allocated Tracks: 2
 Allocated extents: 1

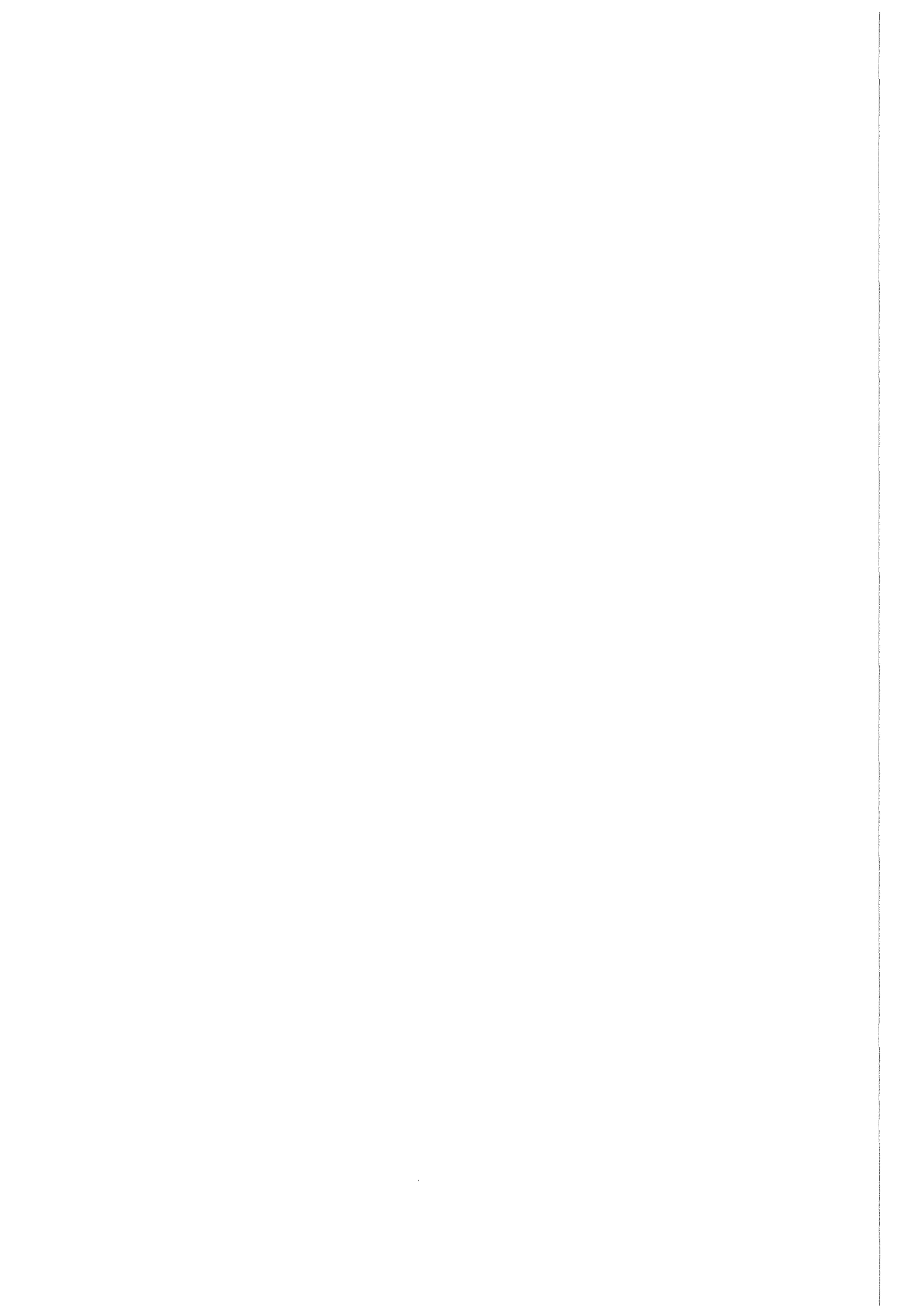
CURRENT UTILIZATION:

Used Tracks: 2
 Used extents: 1

Creation date: 1991/02/13
 Expiration date: ***NONE***

LIBRARY - INR105.MEMORY MEMBERS

NAME	PROCEDURE
ABB1	"FIGUR" & "DDFIG" & "TEXIMA"
ABB2	"FIGUR" & "DDFIG" & "TEXIMA"
ABB3	"FIGUR" & "DDFIG" & "TEXIMA"
ABB4	"FIGUR" & "DDFIG" & "TEXIMA"
BOOKS	"BRAUS" "IMMANI" "PIXI" "PODEL" & "PONUM"
BRAUS	"BRAUS"
CORE	"CORE"
DAMA	"DAMA"
DATAS	"DATAS" & "ENDE" & "FUNKIN"
DELCAT	"DELCAT"
DRUCKT	"DRUCKT"
DRUDA	"DRUDA"
FEJ	"FEJ"
FIGUR	"FIGUR" & "DDFIG"
FORCO	"FORCO"
FUEGE	"FUEGE"
IMMANI	"IMMANI"
KATALOG	"KATALOG"
KOPIER	"KOPIER"
LADDFI	"DDFIG"
LAFIGU	"FIGUR"
LATEXI	"TEXIMA"
LOESCH	"LOESCH"
MAPPER	"TAFEL"
PICOR	"PICOR"
PIXI	"PIXI"
PONUM	"PODEL" & "PONUM"
PROLET	"PROLET"
SERIE	"SERIE"
SERVIN	"FIGUR" & "DDFIG" & "TEXIMA"
SPY	"SPY"
TEXIMA	"TEXIMA"
VERSAT	"TAFEL"



Anhang B. Wie man eine CLIST-Datei an das System anknüpft

SCHRITT I

```
----- USER APPLICATION SELECTION MENU -----
OPTION ==>

1  ISPF/PDF      - ISPF/Program Development Facility (PDF)
AI AID          - Advanced Interactive program Development
B  BIAF         - User Accounting Facility
D  GML          - Formatting with GML/DCF (Release 3.2)
DN GMLNEU       - Formatting with GML/DCF (Starter Set/BookMaster)
E  EARN         - European Academic and Research Network Services
G  GDDM         - Interactive Graphic Application Services
GT GDDMTEST     - Interactive Graphics Application Services
I  INFO        - Information
L  LISTA        - Display TSO Allocation List (LISTA ST)
P  PADSPF      - SNAPAD Version 4
SD SDF          - SPOOL DISPLAY FACILITY
T  FTOOL       - FORTRAN Tools
X  EXIT        - Terminate the ISPF Dialog

Current PROC ==> VSFORT

Press ENTER to start ISPF/PDF, Enter ?option for help, or ? for system info
Press END Key (PF3) to return to the APPLICATION SERVICES Panel M3090
```

Abbildung 4. USER APPLICATION SELECTION MENU

SCHRITT II

```
***** ISPF APPLICATION HANDLING FACILITY (AHF) *****
*****                                     *****
*****                                     *****
*****                                     *****
```

SYSTEM - ISPF 3.1.0 MVS/XA 2.2.3 TSO/E 2.1.0

User-Application and Library Concatenation Services
Version 3.2

OPTION ==>

CPUMDL - M3090

- | | | |
|---|---|-------------------|
| 0 | - Specify ISPF Parameters and set AHF Options | USERID - inr000 |
| 1 | - Execute the Current ISPF/User Application Menu | LOGON - VSFORT |
| 2 | - Invoke Application Administration Services | TERM - 3278 |
| 3 | - Define Additional User Application-Level Libraries (optional) | LUNAME - Mumus999 |
| 4 | - Initialise from Public Application Pool (predefined subset) | |
| 5 | - <u>Define Additional CLIST/REXX Libraries for File SYSPROC Base</u> | |
| 6 | - Display/Print Utilities for Applications | |
| 7 | - Invoke Application Pool Management Services | |
| 8 | - Change User Program Environment (LOGON procedure) | |
| 9 | - Application Mailbox Services | |
| I | - News (information about new or changed public applications) | |
| X | - Terminate the ISPF Dialog using the List/Log Defaults | |

Press END Key (PF3) to terminate ISPF, press ENTER to start your Master Menu

Abbildung 5. ISPF APPLICATION HANDLING FACILITY (AHF)

SCHRITT III

```
----- EDIT/VERIFY TSO/CLIST LIBRARIES -----  
OPTION =====>  
  
The following data sets have been predefined for SYSPROC :  
  
BASIC CLIST CONCATENATION  =====> 'TSOSYS.VSCLIB'  
                           =====> 'TSOSYS.ISRCLIB'  
                           =====> 'TSOSYS.ADICLIB'  
                           =====> 'TSOSYS.CMDPROC.CLIST'  
                           =====> 'TSOSYS.DCQCLIST'  
                           =====> 'TSOSYS.ECF110.DWXCLS1'  
                           =====>  
  
Enter additional data set names, where desired:  
  
GENERAL CLIST/REXX LIBRARY  =====> 'inr000.CLIST'  
EDIT MACRO CLIST LIBRARY   =====>  
REXX LIBRARY (SYSEXEC)     =====>  
  
Press ENTER to add, press END Key (PF3) to return without updating
```

Abbildung 6. Define Additional CLIST/REXX Libraries for File SYSPROC Base