1 Preface

This manual describes the BS2000/OSD system exits. With the aid of these exits it is possible to interrupt the processing of the instruction sequence for a system component or a user program at a given point and execute an instruction sequence written by systems support. Modifications can thus be made to suit the individual requirements of computer center operation.

1.1 Target group

This manual is intended for use in BS2000/OSD systems support and for programmers with extensive knowledge of Assembler and the operating system.

1.2 Summary of contents

A description of the basic mechanism for exits can be found in the chapter “Introduction to system exits”.

A description of all system exits that are currently available can be found in the chapter “Descriptions of the exits”.

The chapter that follows this contains descriptions of the application of event control in exit routines and the system exits used to monitor access to the program libraries.

At the end of the manual you will find a number of chapters containing lists that will make it easier for you to work with the manual.

Readers requiring more in-depth technical information are advised to refer to the “BS2000/OSD Technical Description” manuals ([18] currently available in german only).

README file - changes to the current BS2000 version

Information on functional changes and additions to the current product version described in this manual can be found in the product-specific README file. You will find the README file on your BS2000 computer under the file name SYSRME.BS2CP.140.E. The user ID under which the README file is cataloged can be obtained from your systems support staff. You can ascertain the file name using the following IMON command:

```
/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=BS2CP, LOGICAL-ID=SYSRME.E
```

You can view the README file using the /SHOW-FILE command or an editor, and print it out on a standard printer using the following command:

```
/PRINT-DOCUMENT $userid.SYSRME.BS2CP.140.E,
                LINE-SPACING=*BY-EBCDIC-CONTROL
```

1.3 Changes since the last version of the manual

The following changes have been made since publication of the “System Exits” manual, Edition December 1996 (BS2000/OSD-BC V3.0).

- The DSECTS output has been brought up to the status for BS20000/OSD-BC V5.0.
- Information for systems with SPARC architecture.

1.4 Notational conventions

The following typographical elements are used in this manual:

`Output`	DSECTS and examples are shown in a typewriter font
	For notes on particularly important information

References to publications are shown in the text by abbreviated titles. The full title of each publication to which reference is made is listed under “Related publications” at the back of the manual.

2 Introduction to system exits

A system exit is a point in the operating system or in a TU program where current processing can be interrupted by calling an exit routine. (TU = Task Unprivileged)

An exit routine is a component written, as a rule, by BS2000/OSD systems support and which is inserted in the operating system. It enables a normally non-modifiable operating system function to be modified to suit the special requirements of a particular computer center's operational procedures. Exit routines can add new functions to the standard ones, or they can modify or replace existing functions. System exits are integral components of the operating system.

Exit routines can be loaded and activated or deactivated during a session, i.e. managed dynamically.

Provisions have been made for calling the exit routines in the system functions listed in the table below. Some functions have two exit points:

– a request exit which allows the user's input to be modified or rejected before BS2000/OSD processing
– a return exit which is provided for checking or modifying the processing results from the BS2000 component before the program returns to the requesting task.

The TU exit which can be called from each TU program occupies a special position here. When and how this exit is employed is entirely the responsibility of systems support, in consultation with the computer center customers.

Overview of the exits

Each exit is classified internally by means of an exit number. The following overview shows the exits and the occasions when they are used:

No.	Exit	Type / when used
000	TU exit	
001 002	Termination exits for terminating programs and tasks	request return
003	SHUTDOWN exit	while the session is being terminated
005 006	OPEN member exit for program libraries	request return
007 008	CLOSE member exit for program libraries	request return
010	JOIN command ADD-USER command MODIFY-USER-ATTRIBUTES command	prior to writing the user record to the user catalog
015	Accounting exit	prior to writing an accounting record
020	PSWORD command MODIFY-USER-PROTECTION command	prior to modifying a password in a user record in the user catalog
025 026	SECURE command SECURE-RESOURCE-ALLOCATION command	request return
030	LOGON exit	with the ENTER-JOB, ENTER-PROCEDURE and SET-LOGON-PARAMETERS commands
032	Job parameter exit	with the ENTER-JOB, ENTER-PROCEDURE, MODIFY-JOB and SET-LOGON-PARAMETERS commands for processing job parameters
033	JV exit	during read access to a special job variable
036	BCAM exit	connection setup to system applications
039	OPEN exit	security
040 041	OPEN exit	request return
042 043	CLOSE exit	request return
044 045	Volume swap for tapes	request return

No.	Exit	Type / when used
050 051	CREATE-FILE/ MODIFY-FILE-ATTRIBUTES/ CREATE-FILE-GENERATION/ MODIFY-FILE-GENERATION-SUPPORT/ ADD-FILE-LINK command FILE macro	request return
054	ATTACH library exit for program libraries	after opening a program library
055 056	DELETE member exit for program libraries	request return
057 058	RENAME member exit for program libraries	request return
060 061	DELETE-FILE/ DELETE-FILE-GROUP/ DELETE-FILE-GENERATION/ DELETE-SYSTEM-FILE/ EXPORT-FILE command ERASE macro	request return
070 071	CREATE-FILE/ CREATE-FILE-GENERATION/ CREATE-FILE-GROUP/ MODIFY-FILE-ATTRIBUTES/ MODIFY-FILE-GENERATION-SUPPORT/MODIFY-FILE-GROUP-ATTRIBUTES command CATALOG macro	request return
075	DCAM	request/event-driven
080 081 082 083	CMD (SDF exits) CMD STMT STMT	request = before processing each command return = after analysis of each command by the command decoder request = before analysis of each statement return = after analysis of each statement
090	SPOOL exit	before output of each record via the printer
091	SPOOL header page	prior to output of the header page
094	SPOOL resources routine	conversion of client/server resource
095	SPOOL exit	prior to output of the channel program
096	SPOOL exit	server selection
097	SPOOL exit	control of jobs for file transfer
110	SAT exit	before writing a SAT protocol record

Overview of macros used and the associated standard libraries

The macros listed below can be used to generate the DSECTs required for the exits.The macros are supplied in the specified libraries. The place where they are stored can be changed by system administration.

Macro name	Standard library
$TERMRF, $DOCSYS,	SIPLIB.BS2CP.140
EX005, EX007, EX054, EX055, EX057	SIPLIB.PLAM.033
$SRMSYE	SIPLIB.SRPMNUC.140
NASXIT, NKSEPAR, EX030, EX032, EX050, EX060, EX070, EX080	SYSLIB.BS2CP.140
EX033	SYSLIB.JV.140
YDDEXPL	SYSLIB.DCAM
EX090, EX091, EX092, EX094, EX095, EX096, EX097	SYSLIB.SPOOL.043
EX110	SIPLIB.SATCP.040

2.1 Functional principle of the exit mechanism

The functional principle of the exit mechanism is illustrated in the following diagram:

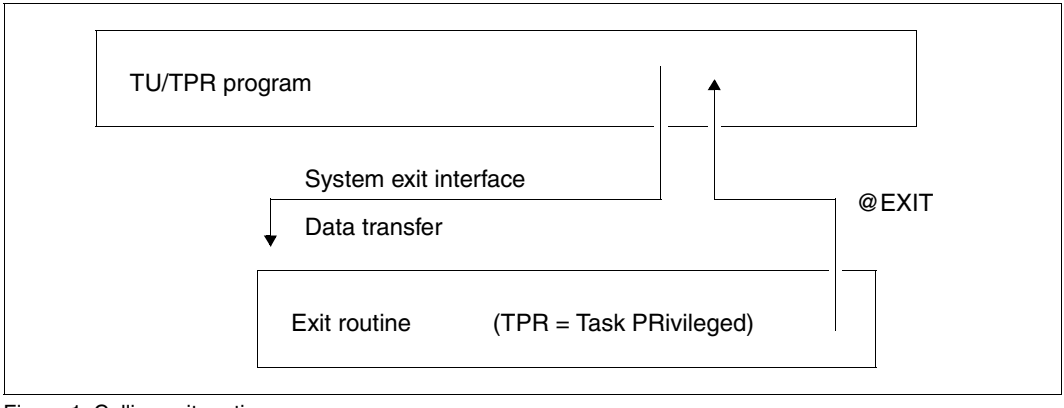


Figure 1: Calling exit routines

Via a TU or TPR macro interface, a module containing an exit point can branch to the corresponding exit routine, provided the routine exists and has been activated. Depending on what is required, this exit routine can check and (if necessary) modify the data, and it can reject or log a request. Control is then returned to the calling system module by means of the @EXIT macro in the program manager environment (DSL environment) or via BR R14 in pure Assembler routines. New exit routines should be developed in the DSL environment wherever possible.

When control branches to an exit routine, certain TPR registers contain information. The register contents should be taken from the description of the relevant exit.

The contents of registers 12, 13 and 14 must not be destroyed by the exit routine; they must have been saved and loaded with the original contents before returning from the exit routine, i.e. they must have their original contents on returning.

Registers 0, 2 and 3 are now only defined for system exits which internally use an old system interface. For exits which have already been converted, registers 0, 2 and 3 are not defined. For details, see the descriptions of the exits.

The contents of the EXVT and the TCB must be accessed via the unbundled access functions (see the “Developers Handbook”).

As stipulated in the relevant exit description, the return code of the exit routine must be stored either in register 15 or in the specified parameter list with a standard header.

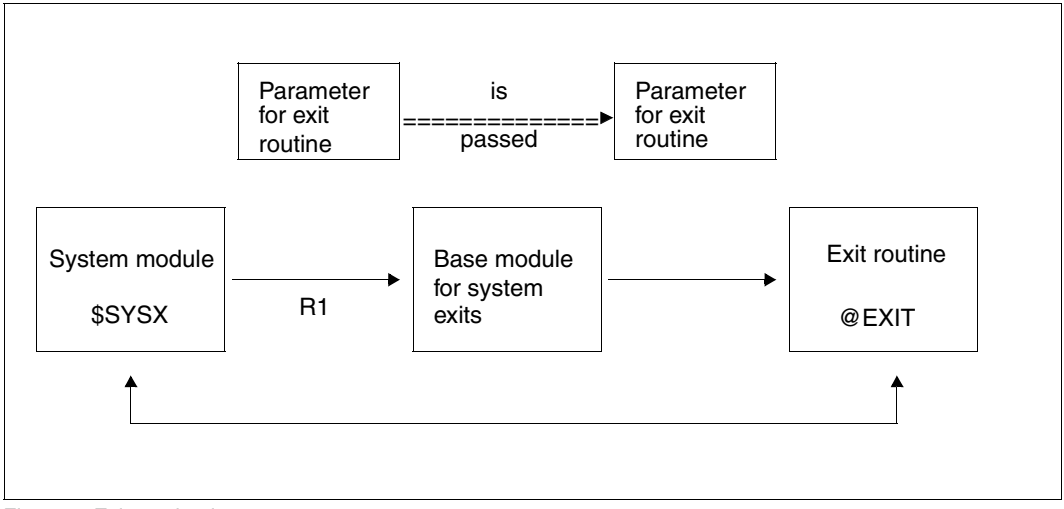


Figure 2: Exit mechanism

2.2 TU macro interface

EXCALL – exit call for TU programs (O type)

The EXCALL macro can be used from within TU programs to call exit routine 000.

[name] EXCALL

The contents of TU registers 0 and 15 are destroyed.

Systems support is responsible for deciding which parameters are passed on to exit routine 000 in register 1, the decision being arrived at in consultation with the caller. The information in register 1 is passed to the TPR PCB from the TU PCB.

The parameters transferred must be validated by the exit routine ($HVAL, $VALID). Care must be taken that these parameters can be accessed.

Otherwise exit routine 000 is governed by the same programming rules as all the other exit routines.

2.3 Exit routines

2.3.1 Programming rules for exit routines

Exit routines, like the system components calling them, always run in the privileged state. Special care should therefore be taken in programming them. The use of TU macros, in particular, or other programming errors may result in the abortion of a task or even in a system crash. Errors in exit routines are treated like system errors. The following rules and restrictions should be observed at all costs:

– Exit routines must include AMODE ANY and RMODE ANY in the XS environment and they must adhere to programming regulations (cf. section “Programming example” on page 14).

– Exit routines must be programmed reentrant and read-only.

– The CSECT/ENTRY name of an exit routine is freely selectable. CSECT/ENTRY names used must be unique throughout the system.

– Any problems arising from call nesting, parallel usage or recursion must be taken care of by the exit routine itself (cf. programming example).

– If both a TU SVC and a privileged interface (e.g. REQM, $REQM) exist for a function, the exit routine must use the privileged call.
Exception: PRNT, cf. note on the spoolout exit.

– Fields marked “unused” or “reserved” in the system tables (see appendix) must not be used by the exit routines. If the exit routine requires task or job information, use of the $TINF, $RDUID, $JINBAS, $JINTSK and JOBINFO macros is strongly recommended. With the aid of these macros the appropriate information can be retrieved or requested from the exit routine’s own parameter areas without any problems.

– Privileged machine instructions cannot be used.

– The exit routine must not be used for system tasks (interrupt-driven tasks); interrogate the appropriate bit in the TCB via $EZOTYP.

– When processing a file in an exit routine, any file that has been opened must also be closed by the same routine, otherwise problems may arise.

– Exit routines run in privileged state TPR (Task PRivileged, previously P2). Exit routines can employ privileged system interfaces *at the user’s risk.*

– Exit routines are activated at a time when no “lock” state exists in the system.

– Exit routines may use the TPR floating-point registers, provided their contents are saved before use and reloaded afterwards.

– When macros (TYPIO, $REQM...) are used, “incapsulation ” macros ##BAL and ##BALR must be diverted to BAS or BASR by means of the Assembler statement OPSYN.

2.3.2 Memory management

Exit routines occupy the class 4 memory of the privileged system address space in BS2000/OSD. They are regarded as a dynamically loadable part of the system and consequently not protected against overwriting.

Task-local memory, e.g. for parameters, can be requested and released in class 5 memory with the aid of the privileged system functions $GETMEM and $RETMEM, respectively.

Global memory must always be reserved as class 4 memory.

Task-local or global memory for the exit routines can be reserved at system generation time with the aid of the class 2 options TASKVECT and SYSVECT; default value: 0 pages; maximum value: 15 pages.

The task vector and system vector are set up after the first call of an exit routine. The task vector is released at LOGOFF or when the task is destroyed; the system vector is retained till shutdown.

The address of the task vector is obtained by calling $TANC with the operands FUNCT=READ and OWNER=SYSEXIT. The address of the system vector is obtained by calling $SANC with the operands FUNCT=READ and OWNER=SYSEXIT. The fields ETCBSEXP (TCB) and EXVTSEXP (EXVT) should not be accessed directly. For further details, see the “Developer’s Handbook ”.

The length of each vector expressed in bytes is stored in the first half-word of the vector. Since bit 2^{15} is set for vector lengths greater than 7 pages, the Assembler instruction LH cannot be used when loading the length in a register.

The system vector and the task vector are used by all exits; access coordination is the responsibility of the exit programmer or systems support.

It is the user’s responsibility to avoid memory bottlenecks caused by the use of task and system vectors or by $REQM or by the exit routines residing in class 4 memory.

2.3.3 Debugging and testing

When an exit routine is called, bit 2^4 of byte ESTKIND is set in the process control block (PCB):

```
ESTKEXIT EQU X'1O'
```

At the end of the exit routine the bit is reset. The names used in the PCB can be generated via the DSTK macro.

Exit routines can be debugged with the products DAMP, HELGA and IDIAS. DAMP is described in the “Diagnostics Handbook” [3]. HELGA and IDIAS are internal diagnostic tools for the service technician.

The settings of AUDIT indicators are not changed by calling an exit routine.

Exit routines are treated as system components when taking dumps; no special measures are performed.

2.3.4 File access monitoring

Using the system exits of the Data Management System for the functions OPEN, FILE, ERASE and CATALOG, systems support can monitor access to files and limit it if necessary. The AUDIT operands in the CREATE-FILE-[GROUP]/MODIFY-FILE[-GROUP]-ATTRIBUTES commands can be used to assign a flag to certain files. This flag is analyzed by the exit routines. For example, a check can be made on how often a certain file was opened or how many files were erased during a task. The exit routine can write the pertinent information temporarily into the task vector. A command or macro can also be rejected. File access monitoring of this type can be limited to certain user IDs and/or certain files.

ADD-USER/MODIFY-USER-ATTRIBUTES command

The *FILE-AUDIT* operand specifies for a user ID whether the user may modify the AUDIT attributes for their files using the CREATE-FILE-[GROUP]/ MODIFY-FILE[-GROUP]-ATTRIBUTES commands or the CATAL macro. Default value: FILE-AUDIT=NO.
Users can check the entry in the user catalog for their user IDs via the SHOW-USER-ATTRIBUTES command.

CREATE-FILE-[GROUP]/MODIFY-FILE[-GROUP]-ATTRIBUTES command

With the aid of the operand *AUDIT=*NONE/*SUCCESS/*FAILURE/*ALL* the user or systems support can specify for each file whether, and in what way, file access is to be monitored by the exit routines.

*NONE	Default value; no monitoring.
*SUCCESS	command/macro processing which was completed successfully is monitored.
*FAILURE	command/macro processing which was terminated with an error is monitored.
*ALL	all command/macro processing is monitored.

The contents of the AUDIT field in the catalog entry can be checked via the SHOW-FILE-ATTRIBUTES command/macro.

Position of the AUDIT indicator in the system tables (see appendix):

CATALOG parameter area:

```
IDKAUDIT      DC      X '00'              AUDIT PARAMETER
IDKNOAUD      EQU     X '00'              AUDIT NOT SPECIFIED
IDKAUDSU      EQU     X '01'              AUDIT=SUCC
IDKAUDFA      EQU     X '02'              AUDIT=FAIL
IDKAUDAL      EQU     X '04'              AUDIT=ALL
IDKAUDNO      EQU     X '08'              AUDIT=NONE
```

2.4 Programming example

2.4.1 System environment for system exits

Exit routines are considered components of the operating system. Since privileged system interfaces are only available in the 31-bit format and do not depend on the addressing mode, it is necessary to adapt the relevant routines. Instead of switching the addressing mode to the appropriate interfaces (a hardware-dependent process), the exit routines must run in the same addressing mode as the system.

Exit routines therefore:

– must be able to run in any addressing mode (AMODE ANY)
– must adhere to the programming regulations valid for the operating system
– must have the attribute RMODE ANY

EXIT routines can access address fields, the contents of which are provided by user programs (e.g. in parameter lists). These fields must be cleaned up by the system exit routines before further processing or before they have to be prepared for 31-bit addressing.

System Exits for systems with SPARC architecture

Like all routines that run as privileged routines, system exit routines must also run directly (native) on the CPU, this means that they must run as SPARC code directly on the SPARC CPU.

If you wish to use system exits, you must ensure that the source code of the system exits is modified and then compiled again.
When doing this, Fujitsu Siemens is able to provide support and advice if required. For information, contact your Service representative.

Converting system exits is an expense that should be taken into account when planning to change over to systems with SPARC architecture.

2.4.2 Concept for the modular structure of exit routines

An exit routine usually incorporates several functions. For example, the LOGON exit comprises: checking the user ID, checking the account number, checking the LOGON password. In addition, further checks are possible via this interface, but since only one routine can be connected to a given system exit and each computer center integrates different functions, the routines are not interchangeable between different computer centers. The individual functions themselves, however, could be interchanged.

For these reasons, it is practical for exit routines to be made up of a base routine together with one or more subroutines.

Base routine

The base routine is a general routine that can be used with all system exits. It does not contain any special service functions and serves merely to append one or more subroutines to a system exit. The subroutines are executed in the desired order. The return code from the subroutine decides whether the other subroutines are called or whether the base routine is terminated.

Subroutine

Each subroutine contains a single, functionally self-contained unit. With a minimum of programming outlay it is possible to compile an individual package for each system exit on the basis of existing subroutines.

2.4.3 Example of a base routine and a subroutine

The choice of Columbus-Assembler for the example is deliberate, as it means that explicit saving of the register contents and reloading can be dispensed with. Users need only make use of the framework presented below and can write the rest of the routines in Assembler or Columbus-Assembler as desired.

The names of the base routine and the subroutine must be unique within the system and must not clash with names in system modules.

Base routine

```
          START
EXITCOD   AMODE  ANY
EXITCOD   RMODE  ANY
EXPARLI   EX090                          DSECT for exit 090
EXITCOD   CSECT
*****************************************************************************
*        MAINROUTINE                                                        *
*        R1 = EXPARLI A(PARAMETER LIST OF EXIT#90                            *
*****************************************************************************
EXIT090   @ENTR  TYP=E,ENV=SPLSPEC,RETURNS=NO
          ENTRY  EXIT090
          USING  EXPARLI,R1              Use layout of paralist of exit 090
*****************************************************************************
*        CALL FIRST SUBROUTINE                                              *
*****************************************************************************
          @PASS  EXTNAME=SUBROUT1        Call Subrout1
          @IF    NZ                      If return code not ok
          LTR    R15,R15
          @THEN
          ST     R15,IEXRC               set return code in exit paralist
          @EXIT                          then return to system
          @BEND
*****************************************************************************
*        CALL SECOND SUBROUTINE                                             *
*****************************************************************************
          @PASS  EXTNAME=SUBROUT2        Call Subrout2
          @IF    NZ                      If return code not ok
          LTR    R15,R15
          @THEN
          ST     R15,IEXRC               set returncode in exit paralist
          @EXIT                          then return to system
          @BEND
```

```
*****************************************************************************
*        CALL FURTHER ROUTINES                                               *
*****************************************************************************
         :
         :
         @PASS  EXTNAME=SUBROUTN          Call Subrout2
         @IF    NZ                        If return code not ok
         LTR    R15,R15
         @THEN
         ST     R15,IEXRC                 set returncode in exit paralist
         @EXIT                            then return to system
         @BEND
*****************************************************************************
*        END OF MAIN ROUTINE                                                 *
*****************************************************************************
         @EXIT
         @END
         END
```

Subroutine

```
SUBEX001 CSECT
         PRINT  GEN
SUBEX001 AMODE  ANY
SUBEX001 RMODE  ANY
EXITPL   EX090
SUBROUTN @ENTR  TYP=I,ENV=SPLSPEC,RETURNS=NO
         ENTRY  SUBROUTN
*****************************************************************************
*        Application-specific instructions                                   *
*****************************************************************************
         L      R15,IADD                  RETURNCODE WEITERREICHEN
         @EXIT   ,                        RETURN TO BASISROUTINE
         @END    ,
         END
```

2.5 Management of exit routines using DSSM

Exit routines are subsystems.

With the aid of the functions provided by the Dynamic SubSystem Management (DSSM) systems support can create, suspend, resume or delete subsystems during the session.

SSCM (Static Subsystem Catalog Management) generates a static subsystem catalog (SSMCAT), in which the subsystem configuration is defined. This subsystem catalog is loaded by DSSM during system initialization. The act of loading transforms the static subsystem catalog into a dynamic subsystem catalog.

During the session the subsystem catalog can be managed by means of the DSSM command ADD-SUBSYSTEM.
Subsystems can be managed dynamically by means of the commands START-SUBSYSTEM, HOLD-SUBSYSTEM, RESUME-SUBSYSTEM, REMOVE-SUBSYSTEM and STOP-SUBSYSTEM.
Systems support can request information on subsystems by means of the SHOW-SUBSYSTEM-STATUS command.
A logging function for diagnostic purposes can be activated by means of the SET-DSSM-OPTIONS command.

If system exit routines contain entry points which are resolved in other subsystems, they must be included in the subsystem declaration by means of the REFERENCE-SUBSYSTEM operand in the SET-SUBSYSTEM-ATTRIBUTES statement.
If system exit routines use privileged interfaces which are linked to a subsystem (e.g. with SPOOL), the subsystem has to be terminated explicitly before shutdown by means of STOP-SUBSYSTEM.

More detailed information on DSSM and the commands can be found in the manuals “Subsystem Management” [17] and “Commands” [8].

Example

The EXJOI010 exit routine is to be declared as a subsystem. Default values are not specified in the statements; the statements are described in the manual “Subsystem Management” [17] :

```
/START-SSCM ————————————————————————————————————————————————————  (1)
//START-CATALOG-CREATION CATALOG-NAME=$TSOS.SSMCAT.NEW ————————————  (2)
. . .
//SET-SUBSYSTEM-ATTRIBUTES -
//   SUBSYSTEM-NAME=EXJOIN(VERSION=01.0),LIBRARY=EXLIB, - ————————  (3)
//   REP-FILE=*NO,DYNAMIC-CHECK-ENTRY=*NO, -
//   CREATION-TIME=*AFTER-SYSTEM-READY, -
//   MEMORY-CLASS=*SYSTEM-GLOBAL(SUBSYSTEM-ACCESS=*SYSTEM), - ———————  (4)
//   SUBSYSTEM-ENTRIES=EXJOI010( - ———————————————————————————————  (5)
//      MODE=SYSTEM-EXIT(NUMBER=10), -
//      CONNECTION-ACCESS=*SYSTEM, -
//      CONNECTION-SCOPE=*FREE),- ————————————————————————————————  (6)
//   LINK-ENTRY=EXJOIN(AUTOLINK=*ALLOWED) ————————————————————————  (7)
. . .
//CHECK-CATALOG
//SAVE-CATALOG
//END
```

(1) The SSCM subsystem to generate a static subsystem catalog (SSMCAT) is called.

(2) The file which is to accept SSMCAT is defined.

(3) The fixed attributes specific to the subsystem are defined. The name of the subsystem (EXJOIN) is freely selectable.

(4) Address space information. The exit routine is loaded in the privileged system address space.

(5) Definition of the entry point (EXJOI010), subsystem type (system exit) and exit number; only privileged programs have access authorization.

(6) In addition, the operand `CONNECTION-SCOPE=*FREE` can be used if the exit routine is not to be locked for unloading by the tasks that are using it.

(7) Control of the linkage editor run during generation of the EXJOIN subsystem.

After system initialization the exit routine declared in this way can be created in the system with the following command:

```
/START-SUBSYSTEM SUBSYSTEM-NAME=EXJOIN
```

The exit routine is placed in the wait state with the following command:

```
/HOLD-SUBSYSTEM SUBSYSTEM-NAME=EXJOIN
```

The suspended exit routine can be reactivated with the following command:

```
/RESUME-SUBSYSTEM UBSYSTEM-NAME=EXJOIN
```

The command

```
/STOP-SUBSYSTEM SUBSYSTEM-NAME=EXJOIN
```

causes deactivation of the exit routine and the release of the reserved memory space.

The HOLD and DELETE functions first lock the exit routine against other calls and then wait until the last caller has left the routine; only then is the required function carried out.

Information on the current status of the exit routine can be requested with the following command:

```
/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=EXJOIN
```

The active version of the exit routine could be replaced by a new version. This depends on the declaration of the new version with attribute `VERSION-EXCHANGE=*ALLOWED`
The command is:

```
/START-SUBSYSTEM SUBSYSTEM-NAME=EXJOIN,
   VERSION=''02.2',VERSION=PARALLELISM=*EXCHANGE-MODE
```

3 Descriptions of the exits

This chapter deals with the currently available exits.

The column headers on each page gives the exit topics and the exit numbers.

The programming rules listed in the “Exit routines” section (see page 10) must be observed. The “Programming example” section (see page 14) should be referred to where necessary.

Termination exits (001/002)

An exit routine can be connected before or after program termination or before task termination.

At the termination request exit control is transferred to the appropriate exit routine before program or task termination (TERM, ABEND, LOGOFF).

At the termination return exit an exit routine is called after a program has ended.

The exit routines can:
– write their own account files
– carry out housekeeping routines
– analyze the global storage area (SYSVECT)
– modify a /LOGOFF/ABEND operand (in the case of task termination).

Request exit (001)

The following information is passed to the exit routine:

```
R1    = A($TERMRF parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

Return exit (002)

The following information is passed to the exit routine:

```
R1    = A($TERMRF parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

The return code for the calling system component must be stored in the ENRTMRET field of the parameter area.

Notes

– General

  The contents of registers 12, 13 and 14 must not be destroyed by the exit routine.

  Calls which lead to further termination processing, e.g. EXEC, LOAD, TERM and $TERM, are illegal.

  It is not possible to reset or abort termination processing.

  The macro TYPIO (with request for an operator response) and CLOSE ALL should not be used in the exit routine.

  If accounting data is being written to the exit routines, the exits should not be deactivated before shutdown as long as user tasks are still present in the system. As shutdown leads to forced task termination, data could be lost.

– File processing

  Files which are managed by exit routines must be created in the system with the operand USER-ACCESS=*ALL-USERS under any user ID. This is because program and task termination are executed under the control of the task in question and therefore of the corresponding user ID. Thus the files must be accessible by all user IDs.

  For the files managed by exit routines all EXLST exits must be made available. Otherwise there is a danger with a request for program termination that the user program will be terminated with the DMS error code of the exit routine.

  The exit routine itself is responsible for the correct closing of files opened in the exit routine.

– Task termination

  The contents of the ENRTLGFC field in the $TERMRF parameter area (which the user can modify) are taken over, provided its value is valid.

  No activation of the exit routines takes place when system tasks or ARCHIVE subtasks are terminated.

– Return exit on program termination

  All files of the user program are already closed (implicit CLOSE ALL via termination processing) at call time.

DSECT

```
         $TERMRF MF=D
         MFCHK MF=D,                                                   C
               SUPPORT=(C,D,L),                                        C
               PREFIX=E,                                               C
               MACID=NRT,                                              C
               DMACID=NRT,                                             C
               DNAME=NRTPL,                                            C
               PARAM=
ENRTPL   DSECT ,
               *,##### PREFIX=E, MACID=NRT #####
         #INTF REFTYPE=REQUEST,                                        C
               INTNAME=NRTPARA,                                        C
               INTCOMP=001
*
* THIS MACRO GENERATES THE DSECT FOR THE PARAMETER LIST WHICH IS
* EXCHANGED BETWEEN THE MODULS NRTCTRL, NRTTERM AND EXTERNAL MODULS.
*
* TO PREVENT ABUSE OF THE GIVEN EQUATES, THE TEST METHOD TO BE USED IS
* INDICATED BY THE ABBREVIATIONS:
*     (TM)  EACH BIT OF THE DATA FIELD HAS A MEANING OF ITS OWN,
*           NO MEANING IS DISPLAYED AS A COMBINATION OF BITS.
*           USE THE "TM"-STATEMENT WHEN TESTING IF A BIT IS SET.
*           THIS ALSO ALLOWS TESTING OF TWO OR MORE DIFFERENT
*           FEATURES AT THE SAME TIME.
*     (CL)  ONLY A CERTAIN COMBINATION OF SET BITS INDICATES A
*           FEATURE. USE THE "CLI"-STATEMENT WHEN TESTING A
*           COMBINATION.
*
         FHDR  MF=(C,ENRT),EQUATES=NO
         DS    0A
ENRTFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
ENRTIFID DS    0A              0   INTERFACE IDENTIFIER
ENRTFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
ENRTFCT  DS    AL1             2   FUNCTION NUMBER
ENRTFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
ENRTRET  DS    0A              4   GENERAL RETURN CODE
ENRTSRET DS    0AL2            4   SUB RETURN CODE
ENRTSR2  DS    AL1             4   SUB RETURN CODE 2
ENRTSR1  DS    AL1             5   SUB RETURN CODE 1
ENRTMRET DS    0AL2            6   MAIN RETURN CODE
```

```
ENRTMR2  DS    AL1             6   MAIN RETURN CODE 2
ENRTMR1  DS    AL1             7   MAIN RETURN CODE 1
ENRTFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
         ORG   ENRTRET
*                                  ******************
ENRTRC   DS    F                   *  RETURN CODE   *
*                                  ******************
*                                  PLEASE USE THE RETURNCODE
*                                  IN THE STANDARDHEADER
*                                  (MAINCODE 1)!
*                                  ------------------------------
*                                  THIS IS THE ONLY PART OF
*                                  NRTPARA THAT MAY BE MANI-    120
*                                  PULATED BY THE CALLED
*                                  ROUTINES
         ORG   ENRTRC
ENRTSCD  DS    XL1               (CL) SECONDARY CODE
ENRTNACT EQU   X'0C'                 NO ACTION PERFORMED
*
         DS    XL2                   UNUSED
*
ENRTRCD  DS    XL1               (CL) RETURN CODE
ENRTSUCC EQU   X'00'                 FUNCT. SUCCESSFULLY PERFORMED
ENRTERR  EQU   X'04'                 FUNCT. FAULTY - TERM. TASK
*
ENRTRES1 DS    3F                    USED ONLY BY NRT - NOT FOR
*                                    EXTERNAL ROUTINES!
*
ENRTWORK DS    A                     GENERAL WORK AREA ADDRESS
*
ENRTOUT  DS    XL1               (CL) OUTPUT PARAMETER FIELD       120
*
ENRTDMPN EQU   0                     CDUMP SHOULD BE NOT CALLED   120
ENRTDMPY EQU   1                     CDUMP CAN BE CALLED          120
ENRTRETN EQU   2                     RETURN TO PROGRAM            120
ENRTCMDP EQU   3                     CALL COMMAND PROCESSING      120
ENRTCONT EQU   4                     CONTINUE TERMINATION         120
*
ENRTDUMP DS    XL1               (TM) DUMP REQUEST                 120
*
ENRTDIAG EQU   X'04'                 DIAGNOSIS DUMP REQUESTED     121
ENRTSDMP EQU   X'02'                 SYSTEM DUMP REQUESTED        121
ENRTUDMP EQU   X'01'                 USER DUMP REQUESTED          121
ENRTDPNN EQU   X'00'                 NO DUMP REQUESTED            121
*
         DS    XL2                   RESERVED                     120
*
```

```
ENRTINFO DS    A                        SPECIFIC INFORMATION FOR THE
*                                       CALLED ROUTINE. CURRENTLY ONLY
*                                       NAME MANAGER INFORMATION IS
*                                       PROVIDED WHEN REQUIRED.
         ORG   ENRTINFO
ENRTNMCL DS    XL3                      NAME MANAGER INFORMATION FIELD.
ENRTCLAS DS    AL1                (CL) INDICATES THE CURRENT NAME
*                                       MANAGER CLASS OR ZERO FOR
*                                       DYNAMIC CLASSES.
*
*
ENRTREQ  DS    XL1                (TM) TERMINATION REQUEST.
*
ENRTNPTR EQU   X'01'                    NONPRIVILEGED TERMINATION
*                                       REQUEST (TERM,TERMD,TERMJ,TRMJD)
ENRTTERM EQU   X'02'                    PRIVILEGED TERMINATION REQUEST
*                                       ($TERM) AND NONPRIVILEGED
*                                       REQUEST ISSUED IN P2
ENRTTRMT EQU   X'04'                    TERMINATION REQUEST DUE TO
*                                       SYSTEM ERROR                120
*
*
ENRTFUNT DS    XL1                (CL) TERMINATION UNIT TO BE
*                                       EXECUTED.
ENRTPRGR EQU   X'01'                    TERMINATION UNIT IS
*                                       PROGRAM.
ENRTSTEP EQU   X'02'                    TERMINATION UNIT IS STEP.
ENRTTASK EQU   X'04'                    TERMINATION UNIT IS TASK.
ENRTJOB  EQU   ENRTTASK                  TERMINATION UNIT IS JOB.
*                                       AT PRESENT JOB IS EQUIVALENT
*                                       TO TASK
*
*
ENRTTTYP DS    XL1                (CL) TERMINATION TYP
*
ENRTNORT EQU   X'00'                    NORMAL TERMINATION
ENRTABNT EQU   X'04'                    ABNORMAL TERMINATION
*
*
ENRTCUNT DS    XL1                (CL) CURRENT TERMINATION UNIT.
*
*                                       USE THE EQUATES DEFINED UNDER
*                                       "TERMINATION UNIT TO BE
*                                       EXECUTED" FOR PROGRAM, JOB
*                                       STEP, TASK AND JOB TERMINATION!
*
```

```
         DS    0F
ENRTCMPC DS    CL4                    COMPLETION CODE FOR JOB VARIABLE
*                                     SUPPORT
*
ENRTCMD  DS    XL1                (TM) INFORMATION ON THE COMMAND THAT
*                                     INITIATED TERMINATION
ENRTLGOF EQU   X'01'                  LOGOFF COMMAND
ENRTABND EQU   X'02'                  ABEND COMMAND
ENRTCAN  EQU   X'04'                  CANCEL COMMAND
ENRTSHUT EQU   X'08'                  SHUTDOWN COMMAND
ENRTEXLD EQU   X'10'                  EXEC OR LOAD COMMAND
ENRTPOSX EQU   X'20'                  EVENT=POSIX                    121
*
*
ENRTLGFC DS    XL1                (TM) INFORMATION ON LOGOFF COMMAND.
*
ENRTLGBT EQU   X'01'                  LOGOFF BUT
ENRTLGTP EQU   X'02'                  LOGOFF TAPE
ENRTLGNS EQU   X'10'                  LOGOFF NOSPOOL
ENRTLGNO EQU   X'06'                  EXIT-JOB  - NO SYSOUT          201
ENRTLGNL EQU   X'0A'                  EXIT-JOB  - NO SYSLST          201
*
*
ENRTCANC DS    XL1                (TM) INFORMATION ON CANCEL COMMAND
*
ENRTOPER EQU   X'02'                  CANCEL BY OPERATOR-TASK
ENRTTSOS EQU   X'04'                  CANCEL BY $TSOS
ENRTUSID EQU   X'08'                  CANCEL BY SAME USER-ID
ENRTHOST EQU   X'10'                  CANCEL BY ANOTHER HOST
*
*
         DS    CL1                    RESERVED                       120
*
*
*
         DS    XL1
ENRTTRMC DS    CL7                    TERMINATION CODE - 7 BYTES
         ORG   ENRTTRMC+3
ENRTCODE DS    CL4                    TERMINATION CODE - 4 BYTES
*                                     NOT SUPPORTED AFTER V8.0!
*
*
ENRTPLE  DS    0F                     WORD ALIGNMENT IS SUFFICIENT 120
*
ENRT#    EQU   *-ENRTFHE
```

SHUTDOWN exit (003)

The SHUTDOWN exit is activated while a session is terminating.

The corresponding exit routine can, for example, write its own accounting data, record statistical values, etc.

The following information is passed to the exit routine:

```
R1    = A(parameter area $SHUIND)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

Notes

– The contents of registers 12, 13 and 14 must not be destroyed by the exit routine.

– The SHUTDOWN exit routine runs under the control of the operator task. Consequently all program errors lead to abnormal termination of the system.

– Before the SHUTDOWN exit routine is activated a termination request exit routine is activated, if available, for every task that has to be terminated.

OPEN member exits (005/006)

The opening of members of a program library can be monitored by systems support with the aid of two exit points:

– At the request exit systems support can reject the OPEN call before the OPEN operation is performed.

– At the return exit the member has already been opened and the full identification of the member (type, name, version, variant) is known. At the request exit the member identification may still be incomplete due to the search level option, so the return exit offers another chance to reject the OPEN call. The member is then returned to its original state.

Request exit (005)

The following information is passed to the exit routines:

```
R1    = A(exit parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

(cf. chapter “System exits for program libraries” on page 149)

Return code in register 15:

```
R15 =    X'000000RR'

         RR =   Return code of exit routine
                00  no error
                04  OPEN member rejected
```

Return code in the exit parameter area:
In addition to the return code in register 15, the following fields are also provided: PMOPM (Open Modus), PMSLV (Search Level) and PMLVD (Search Default).

Return exit (006)

The return exit is identical to the request exit.

Notes

– The search level (PMSLV) is the hierarchy level within the member identification. As only one variant is currently permitted, “NO LEVEL” and “VARIANT” are identical in meaning.

– “SEARCH LEVEL = NO LEVEL” indicates that a member for which all entries have been made is expected.

– “SEARCH DEFAULT” (PMLVD) specifies whether the first or the last member on the specified search level is to be opened.

DSECT

The DSECT for the parameter area can be generated by means of the EX005 macro:

```
name EX005 D[,[prefix],[ALL]]
```

```
         EX005 D
IEX005   DSECT
         DS    0F
*****************************************************
* OPEN    PARAMETER LIST FOR RZ ROUTINES            *
*****************************************************
PMRC     DC    F'0'                    PLAM RETURN CODE
PMRC1    DC    F'0'                    PLAM SECONDARY RETURN CODE
PMPLV    DC    CL8' '                  VERSION OF PLAM: VZZ.ZBZZ
PMLIB    DC    A(0)                    A(LIBRARY NAME):   CL54
PMLINK   DC    A(0)                    A(LIBRARY NAME LINK):   CL8
PMOPM    DC    CL1' '                  OPEN MODUS: I: INPUT
*                                                  W: WRITE
*                                                  U: UPDATE
PMSLV    DC    CL1' '                  SEARCH LEVEL: L: LIBRARY
*                                                    T: TYPE
*                                                    N: NAME
*                                                    V: VERSION
*                                                    R: VARIANT
*                                                    O: NO LEVEL
PMLVD    DC    CL1' '                  SEARCH DEFAULT: N: NO
*                                                      H: HIGH
*                                                      L: LOW
         DC    CL1' '
PMTYP    DC    A(0)                    A(MEMBER TYPE):    CL8
PMNAM    DC    A(0)                    A(MEMBER NAME):    CL64
PMVER    DC    A(0)                    A(MEMBER VERSION): CL24
PMVAR    DC    A(0)                    A(MEMBER VARIANT): F
PMDAT    DC    A(0)                    A(MODIFICATION DATE): CL14: DATE
*                                                              ,CL8:  TIME
         DC    A(0)
         DC    A(0)
         DC    A(0)
LEN      EQU   *-PMRC
```

CLOSE member exits (007/008)

The closing of program library members can be monitored by means of two exit points:

– At the request exit systems support can obtain information on the member, in the same way as via the return exit of the OPEN member. The CLOSE member cannot be prevented.

– At the return exit the member has already been closed. Here systems support receives the full identification of the member, including the date on which members were generated or modified.

Request exit (007)

The following information is passed to the exit routines:

```
R1    = A(exit parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

(cf. chapter “System exits for program libraries” on page 149)

Return code:

```
R15 =    X'000000RR'

         RR =   Return code of exit routine
                00  no error
                04  CLOSE member rejected (ignored by PLAM)
```

In addition to the return code in register 15, the PMCLM field is also filled.

Return exit (008)

The return exit is identical to the request exit.

DSECT

The DSECT for the parameter area can be generated by means of the EX007 macro:
`name EX007 D[,[prefix],[ALL]]`

```
         EX007 D
IEX007   DSECT
         DS    0F
*****************************************************
* CLOSE   PARAMETER LIST FOR PLAM RZ ROUTINES       *
*****************************************************
PMRC     DC    F'0'                    PLAM RETURN CODE
PMRC1    DC    F'0'                    PLAM SECONDARY RETURN CODE
PMPLV    DC    CL8' '                  VERSION OF PLAM: VZZ.ZBZZ
PMLIB    DC    A(0)                    A(LIBRARY NAME):   CL54
PMLINK   DC    A(0)                    A(LIBRARY NAME LINK):   CL8
         DC    CL1' '
         DC    CL1' '
         DC    CL1' '
PMCLM    DC    CL1' '                  CLOSE MODE: Y: KEEP
*                                                  N: DELETE
PMTYP    DC    A(0)                    A(MEMBER TYPE):    CL8
PMNAM    DC    A(0)                    A(MEMBER NAME):    CL64
PMVER    DC    A(0)                    A(MEMBER VERSION): CL24
PMVAR    DC    A(0)                    A(MEMBER VARIANT): F
PMDAT    DC    A(0)                    A(MODIFICATION DATE): CL14: DATE
*                                                              ,CL8:  TIME
         DC    A(0)
         DC    A(0)
         DC    A(0)
LEN      EQU   *-PMRC
```

JOIN exit (010)

The JOIN exit routine can be used to accept or reject an /ADD-USER or /MODIFY-USER-ATTRIBUTES command. The information required to make the corresponding decision is made available in a variable change list. This list contains the change requests issued by the command in the form of a variable list. The structure of this list is similar for the addition of a new user ID (ADD-USER) or the modification of attributes of existing user IDs (MODIFY-USER-ATTRIBUTES). The address of the variable change list is passed in the exit parameter area $SRMSYE after the standard header in the SRMSYEPT field. Additional information on the relevant user IDs or on other system values can be queried using the appropriate system interfaces.

The following information is passed to the exit routine:

```
R1    = A(parameter area $SRMSYE)
R2 .... R11 are not defined.
R12   = A(TPR program manager)
R13   = A(save area)
R14   = A(indirect return)
R15   = A(exit routine)
```

The format of the entries in the change list is described in the SJMCHKZ macro. By studying the complete list it is possible to work out what operands were specified for the relevant command.

Note

The new password is always supplied at the interface in unencrypted form.

The **change list** for the **ADD-USER command** has the following format:

```
User ID of command originator          8 bytes
User identifier                        X'01'
User ID to be processed                8 bytes
Group identifier                       X'02'
Group ID                               8 bytes
:                                      :
:                                      :
End identifier                         X'FF'
```

The list continues with values which are not included in the user record but which may be used for evaluation purposes.

```
Encrypt identifier                  1 byte   "Encrypt identifier"
Encrypt                             1 byte   Encrypt value
                                             01 – yes
                                             02 – no
Pubset identifier                   1 byte   "Pubset identifier"
Pubset ID                           4 bytes  Pubset (possibly # for home)
```

The **change list** for the **MODIFY-USER-ATTRIBUTES command** has the following format:

```
User ID of command originator       8 bytes
User identifier                     X'01'
User ID to be processed             8 bytes
:                                   :
Modification identifier             X'xx'
Modification date                   Number of bytes = length of
                                    modification date
:                                   :
:                                   :
End identifier                      X'FF'
```

The list continues with values which are not included in the user record but which may be used for evaluation purposes.

```
Encrypt identifier                  1 byte   "Encrypt identifier"
Encrypt                             1 byte   Encrypt value
                                             01 – yes
                                             02 – no
Pubset identifier                   1 byte   "Pubset identifier"
Pubset ID                           4 bytes  Pubset (possibly # for home)
```

Meaning

“Encrypt” indicates whether the specified password is to be added to the user record in encrypted form (YES/01) or unencrypted form (NO/02).

The pubset identifier indicates which pubset is to be affected by the command.

Format of combined entries:

Some entries in the modification list consist not just of a single value (the meaning of which is apparent), but of a combination of values. These are:

Identifier		Field	Length
Name	Value		
PASS	X'03'	Encryption (1=yes, 0=no) New identifier -------- empty --------	1 byte 8 bytes 8 bytes
TESTPRIV	X'10'	READ privilege WRITE privilege CONTROLLED indicator	1 byte 1 byte 1 byte
USS	X'15'	Switch ON Switch OFF Switch value	4 bytes 4 bytes 4 bytes

If a MODIFY-USER-ATTRIBUTES command is issued, account number data is saved more than once. This is because the account number has to be created or, if it already exists, the related data has to be modified. This leads to repetitions in the variable list.

The account numbers are therefore stored in the list as follows:
– If the account number is new, a full list of values is saved.
– If the account number already exists, the values specified are appended as a list.

Full list of values for an account number

```
Account identifier                    X'1A'
Account number                        8 bytes'
NTL identifier                        x'1B'
NTL value                             1 byte
:                                     :
:                                     :
CLASS identifier                      X'21'
CLASS value                           1 byte
```

Values actually specified for an account number

```
:                                     :
TIME identifier                       X'1F'
TIME value:                           4 bytes
:                                     :
```

The lists can be analyzed with the aid of the SJMCHKZ macro.

The return code for the calling system component is supplied via the maincode of the $SRMSYE parameter area in the two rightmost bytes of the SRMSYEHDR field.

```
SRMMR1=X'00'          The command is to be executed.
SRMMR1=X'01'          The command is to be rejected.
```

If the exit routine has rejected the ADD-USER-/MODIFY-USER-ATTRIBUTES command, the following message is issued:

```
% SRM2108 COMMAND REJECTED BY SYSTEM EXIT ROUTINE
```

DSECT

```
         $SRMSYE MF=D
         MFTST MF=D,PREFIX=S,MACID=RMF,ALIGN=F,                          C
               DMACID=RMF,SUPPORT=(D,C,M,L),DNAME=RMF_MDL
SRMF_MDL DSECT ,
               *,##### PREFIX=S, MACID=RMF #####
*   parameterarea description
SRMFSYEHDR FHDR MF=(C,SRMF),EQUATES=NO                  Standardheader
SRMFSYEHDR DS    0A
SRMFFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
SRMFIFID DS    0A              0   INTERFACE IDENTIFIER
SRMFFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
SRMFFCT  DS    AL1             2   FUNCTION NUMBER
SRMFFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
SRMFRET  DS    0A              4   GENERAL RETURN CODE
SRMFSRET DS    0AL2            4   SUB RETURN CODE
SRMFSR2  DS    AL1             4   SUB RETURN CODE 2
SRMFSR1  DS    AL1             5   SUB RETURN CODE 1
SRMFMRET DS    0AL2            6   MAIN RETURN CODE
SRMFMR2  DS    AL1             6   MAIN RETURN CODE 2
SRMFMR1  DS    AL1             7   MAIN RETURN CODE 1
SRMFFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
SRMFSUCC  EQU   0                       CMD accepted
SRMFREJE  EQU   1                       CMD rejected
*
SRMFSYEPT DS    A                       POINTER
SRMF#     EQU   *-SRMFSYEHDR
```

ACCOUNTING exit (015)

An accounting exit routine that can be connected by systems support is called for every accounting record before the record is transferred to the accounting write task and written in the accounting file.

The exit routine can accept the accounting record unchanged, modify it, suppress it or create additional accounting records, which can be inserted before or after the current record in the accounting file.

The following information is transferred to the exit routine:

```
R1    = A(parameter area NASXIT)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

The return code for the calling system component is transferred in the NASXRET field of the parameter area:

– modify the accounting record (return code X'00')

  The address of the original accounting record is transferred to the exit routine by way of the parameter list. The routine can make any changes to the record, provided the record length does not exceed 496 bytes. The layout of system accounting records is described in detail in the manual "Introductory Guide to Systems Support" [7].

– suppress the accounting record (return code X'08')

  Apart from setting return code X'08', no action is required as a result of the exit routine.

– insert an accounting record (return code X'00')

  The records to be inserted must be written to the accounting file by the exit routine with the $NAREC macro before returning to the system. The record identifiers of customer-specific accounting records should begin with one of the letters X, Y or Z in order to avoid name conflicts with the system's name identifiers.

– add an accounting record (return code X'04')

  The system writes the accounting record into the accounting file and then returns control to the exit routine. In this case the NASXRECA field of the parameter list contains no address. The exit routine can then add more accounting records with $NAREC. If control is returned to the system again, the return code is not interpreted. A third call is not provided for.

– Excluding the possibility of recursion

  The accounting exit routine is also called for accounting records, which are written by the routine itself with $NAREC. A possible recursion must therefore be prevented within the exit routine.

DSECT

The NASXIT macro generates the layout of the exit parameter area as a DSECT or a data area.

The exit parameter area contains the standard header (including the return code) and the NASRECA field for the address of the accounting record which is to be added to the accounting file.

The exit routine's return code must be added to the NASREXT field ("General Return Code") of the parameter area.

```
          NASXIT MF=D
          MFCHK MF=D,                                                  C
                SUPPORT=(C,D),                                         C
                PREFIX=N,                                              C
                MACID=ASX,                                             C
                DMACID=ASX,                                            C
                DNAME=ASXIT
NASXIT    DSECT ,
                *,##### PREFIX=N, MACID=ASX #####
          #INTF INTNAME=$NASXIT,REFTYPE=REQUEST,INTCOMP=1
* PARAMETER LIST LAYOUT
NASXACC   FHDR MF=(C,NASX),EQUATES=NO
NASXACC   DS    0A
NASXFHE   DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
NASXIFID DS     0A              0   INTERFACE IDENTIFIER
NASXFCTU DS     AL2             0   FUNCTION UNIT NUMBER
*                                   BIT 15    HEADER FLAG BIT,
*                                   MUST BE RESET UNTIL FURTHER NOTICE
*                                   BIT 14-12 UNUSED, MUST BE RESET
*                                   BIT 11-0  REAL FUNCTION UNIT NUMBER
NASXFCT   DS    AL1             2   FUNCTION NUMBER
NASXFCTV DS     AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
NASXRET   DS    0A              4   GENERAL RETURN CODE
NASXSRET DS     0AL2            4   SUB RETURN CODE
NASXSR2   DS    AL1             4   SUB RETURN CODE 2
NASXSR1   DS    AL1             5   SUB RETURN CODE 1
NASXMRET DS     0AL2            6   MAIN RETURN CODE
NASXMR2   DS    AL1             6   MAIN RETURN CODE 2
```

```
NASXMR1  DS    AL1             7   MAIN RETURN CODE 1
NASXFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*
* FOLLOWING RETURN CODES MAY BE SET :
*
*    SC2   SC1   MAIN         MEANING
*     00    00   0000         WRITE ACC RECORD AND DO NOT RECALL EXIT
*     00    00   0004         WRITE ACC RECORD AND RECALL EXIT
*     00    00   0008         DO NOT WRITE ACC RECORD
*
* EQUATES FOR SUBCODE1 :
*
NASXS1OK EQU  X'00'                    NORMAL PROCESSING
*
* EQUATES FOR MAINCODE :
*
NASXMWR  EQU  X'00'                    WRITE ACC RECORD
NASXMWRR EQU  X'04'                    WRITE ACC RECORD AND RECALL
NASXMNWR EQU  X'08'                    DO NOT WRITE THE RECORD
*
NASXRECA DS    A             ACC RECORD ADDRESS
*
* EQUATE FOR LENGTH OF PARAMETERLIST :
*
NASX#    EQU  *-NASXACC
```

PSWORD exit (020)

A PSWORD exit routine can accept or reject a MODIFY-USER-PROTECTION command. The data required for checking purposes is supplied in a variable change list. The address of this list is passed in the exit parameter area $SRMSYE after the standard header in the SRMFSYEPT field.
Additional information on the relevant user IDs or on other system values can be queried using the appropriate system interfaces.

The following information is passed to the exit routine:

```
R1    = A(parameter area $SRMSYE)
R2 .... R11 are undefined.
R12   = A(TPR program manager)
R13   = A(save area)
R14   = A(indirect return)
R15   = A(exit routine)
```

The exit routine can be called in two different formats. They can be distinguished by the SRMFFCTV version code (see DSECT on page 42).

SRMFFCTV=1

This format is used to call the exit routine when processing the MODIFY-USER-PROTECTION command, when SECOS version V3.0 or higher is in use. As a result, there are two calls in this case:

1. A change list is produced, showing whether it was established in the exit routine, if the command should be carried out or not.

   The change list has the following format:

```
Identifier for password modification   X'03'
------- not used -------               X'01'
New password (uncoded)*                8 bytes
Old password*                          8 bytes
End identifier                         X'FF'
PVS flag                               1 byte
PVS ID                                 4 bytes
User ID                                8 bytes
TID                                    4 bytes
RUN (first or second call)             1 byte
RESULT(result of the call)             1 byte
```

   * The new password is always provided in unencrypted form at the interface. The old password is encrypted in accordance with the class 2 option ENCRYPT.

In the case of this call, the RUN field has the value 1 and the RESULT field has the value 0.

The return code for the calling system component is supplied via maincode1 of the $SRMSYE parameter area in the rightmost byte of the SRMFSYEHDR field.

```
SRMFMR1=X'00'        The command is to be executed.
SRMFMR1=X'01'        The command is to be rejected.
```

Subsequently, further checks are carried out in the system to establish whether the password can be changed or not, e.g. a check on sufficient complexity.

2. The RESULT field of the change list shows whether or not the password could be successfully changed:

```
RESULT=1             The password has been successfully changed.
RESULT=2             The password has not been changed.
```

The RUN field has the value 2.

If the exit routine has rejected the MODIFY-USER-PROTECTION command, the following message is issued:

```
% SRM2301 COMMAND REJECTED BY SYSTEM EXIT ROUTINE
```

SRMFFCTV=0

This format is used to call the exit routine when processing the MODIFY-USER-PROTECTION command, either when SECOS is not in use or when a version earlier than V3.0 is being used. When processing the ISP commando, PSWORD, the exit routine is called in this format in any case, regardless of whether SECOS is being used, and which version.

The change list has the following format:

```
Identifier for password modification   X'03'
------- not used -------               X'01'
New password (uncoded)*                8 bytes
Old password*                          8 bytes
End identifier                         X'FF'
PVS flag                               1 byte
PVS ID                                 4 bytes
User ID                                8 bytes
```

* The new password is always provided in unencrypted form at the interface. The old password is encrypted in accordance with the class 2 option ENCRYPT.

The return code for the calling system component is supplied via maincode1 of the $SRMSYE parameter area in the rightmost byte of the SRMFSYEHDR field.

```
SRMFMR1=X'00'        The command is to be executed.
SRMFMR1=X'01'        The command is to be rejected.
```

If the exit routine has rejected the MODIFY-USER-PROTECTION command, the following message is issued:

```
% SRM2301 COMMAND REJECTED BY SYSTEM EXIT ROUTINE
```

DSECT

```
         $SRMSYE MF=D
         MFTST MF=D,PREFIX=S,MACID=RMF,ALIGN=F,                          C
               DMACID=RMF,SUPPORT=(D,C,M,L),DNAME=RMF_MDL
SRMF_MDL DSECT ,
               *,##### PREFIX=S, MACID=RMF #####
*   parameterarea description
SRMFSYEHDR FHDR MF=(C,SRMF),EQUATES=NO                  Standardheader
SRMFSYEHDR DS    0A
SRMFFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
SRMFIFID DS    0A              0   INTERFACE IDENTIFIER
SRMFFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
SRMFFCT  DS    AL1             2   FUNCTION NUMBER
SRMFFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
SRMFRET  DS    0A              4   GENERAL RETURN CODE
SRMFSRET DS    0AL2            4   SUB RETURN CODE
SRMFSR2  DS    AL1             4   SUB RETURN CODE 2
SRMFSR1  DS    AL1             5   SUB RETURN CODE 1
SRMFMRET DS    0AL2            6   MAIN RETURN CODE
SRMFMR2  DS    AL1             6   MAIN RETURN CODE 2
SRMFMR1  DS    AL1             7   MAIN RETURN CODE 1
SRMFFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
SRMFSUCC  EQU   0                       CMD accepted
SRMFREJE  EQU   1                       CMD rejected
*
SRMFSYEPT DS    A                       POINTER
SRMF#     EQU   *-SRMFSYEHDR
```

SECURE exits (025/026)

The application of the SECURE-RESOURCE-ALLOCATION command can be monitored by systems support at two exit points:

– The request exit routine can reject the SECURE request.

– After the SECURE request is completed, the return exit routine can lead to suitable actions being logged or registered.

Return exit (025)

The following information is passed to the exit routine:

```
R1    = A(NKSEPAR parameter area)
R12   = A(P2 program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

Notes

– The contents of registers 12, 13 and 14 must not be destroyed by the exit routine.

– The NKSHMSG field of the NKSEPAR work area contains:
  – the message number (_NKSxxxx) with which the SECURE-RESOURCE-ALLOCATION command was rejected by the command processor
  – XL8’00’ for successful command processing.

– The NKSHPTR field of the NKSEPAR work area contains:
  – a pointer to the erroneous element of the NKSEPAR data list or, if the erroneous element cannot be determined easily, A(0)
  – A(0) for successful command processing.

Request exit (026)

The register values are supplied in the same way as for the return exit.

The return code for the calling system component is passed in the NKSHEXR field of the NKSEPAR parameter area.

```
NKSHEXRC=X'00'  /SECURE-RESOURCE-ALLOCATION was successful.
NKSHEXRC=X'04'  /SECURE-RESOURCE-ALLOCATION was rejected by the  exit routine.
```

Notes

– The contents of registers 12, 13 and 14 must not be destroyed by the exit routine.

– If the SECURE request is rejected by the exit routine, the following message is displayed:

```
%  NKS0062 SECURE REQUEST REJECTED BY THE SYSTEM ADMINISTRATOR.
           REASON: (&00)
```

One possible reason for the rejection could be that more than three devices of type 3590 are to be reserved.

– If the SECURE-RESOURCE-ALLOCATION command is rejected by the exit routine (NKSHEXRC = X'04'), the NKSHPTR field of the NKSEPAR work area can be supplied with the start address of the errored element in the NKSEPAR data list. The user then receives further information via message NKS0062 (see above).

DSECT

A DSECT for NKSEPAR (the address is transferred in register 1) can be generated with the NKSEPAR macro:

```
         NKSEPAR
         MFCHK DNAME=KSHEAD,MF=D,MACID=KSH,ALIGN=F,PREFIX=N,        200C
               DMACID=KSH                                           200
NKSHEAD  DSECT ,
               *,##### PREFIX=N, MACID=KSH #####
*********************************************************
**     SYSTEM-EXIT FOR SECURE : HEADER DESCRIPTION      **
*********************************************************
*--                            1 SECURE_EXIT_HEADER,
NKSHEXRT DS    OF               2 EXIT_RETURN_INFO                  200
         DS    XL3                FILLER                            200
NKSHEXRC DS    X                  EXIT_RETURN_CODE                  200
NKSHEXOK EQU   X'00'              SUCCESS                           200
NKSHEXSR EQU   X'04'              SEC_RES REJECTED                  200
NKSHMSG  DS    XL8              2 RETURN_EXIT_MSG_IDENTIFIER
*--                               (SEC_SUCCESS = XL8'00',
*--                                SEC_NO_SUCCESS = ' NKS....'),
NKSHPTR  DS    A                2 SECURE_EXIT_ERROR_ELEMENT
*--                               (DEFINED = A(BEGIN_OF_ERROR_ELEM,
*--                                NOT_DEFINED = A(0));
*--                            2 POINTERS_AND_COUNTERS_OF_ELEMENTS
*--                               (UNDEFINED=POINTER_AND_COUNTER_
*--                                                    _ARE_SET_TO_ZERO)
NKSHFILE DS    A                  3 FILE_LIST_POINTER,
NKSHFIL# DS    H                  3 FILE_ELEM_COUNTER,
NKSHVOL  DS    A                  3 VOLUME_LIST_POINTER,
NKSHVOL# DS    H                  3 VOLUME_ELEM_COUNTER,
NKSHDEV  DS    A                  3 DEVICE_LIST_POINTER,
NKSHDEV# DS    H                  3 DEVICE_ELEM_COUNTER,
NKSHUNIT DS    A                  3 UNIT_LIST_POINTER,
NKSHUNI# DS    H                  3 UNIT_ELEM_COUNTER,
NKSHDEVL DS    A                  3 DEVICE_AT_LOCATION_LIST_POINTER 102
NKSHDE#L DS    H                  3 DEVICE_AT_LOCATION_ELEM_COUNTER 102
         DS    OF
****       ELEMENT_DESCRIPTION SEE BELOW           ****
NKSHLEN  EQU   *-NKSHEXRT       LENGTH_OF_HEADER;
```

```
         SPACE 2
         MFCHK DNAME=KSELEM,MF=D,MACID=KSE,ALIGN=F,PREFIX=N,        200C
               DMACID=KSE                                           200
NKSELEM  DSECT ,
               *,##### PREFIX=N, MACID=KSE #####
*********************************************************
**     SYSTEM-EXIT FOR SECURE : ELEMENT DESCRIPTION     **
*********************************************************
*--                            1 SECURE_EXIT_ELEMENTS ALIGN FW,
NKSETYP  DS    XL1               2 TYPE_OF_ELEMENT SET
NKSEFILE EQU   X'01'                (FILE   = 1,
NKSEVOL  EQU   X'02'                 VOLUME = 2,
NKSEDEV  EQU   X'03'                 DEVICE = 3,
NKSEUNIT EQU   X'04'                 UNIT   = 4,
NKSEDEVL EQU   X'05'                 DEVICE AT LOCATION = 5),       102
         SPACE 1
NKSEDESC DS    0C                2 ELEMENTS_DESCRIPTION,
*** LAYOUT FOR FILE ELEMENT ***   3 FILE_ELEMENT,
NKSEFNAM DS    CL54                4 FILE_NAME,
NKSEFTYP DS    XL1                 4 FILE_DEVICE_TYPE SET
NKSEFTDI EQU   X'01'                 (DEVICE_TYPE_DISK = 1,
NKSEFTTA EQU   X'02'                  DEVICE_TYPE_TAPE = 2),
NKSEFRES DS    XL1                 4 FILE_RESERVATION_TYPE SET
NKSEFRSH EQU   X'00'                 (SHAREABLE = 0,
NKSEFREX EQU   X'01'                  EXCLUSIVE = 1),
NKSEFACC DS    XL1                 4 FILE_ACCESS_IF_TAPE SET
NKSEFARD EQU   X'00'                 (READ  = 0,
NKSEFAWR EQU   X'01'                  WRITE = 1),
*
NKSEFLEN EQU   *-NKSETYP          LENGTH_OF_ONE_FILE_ELEMENT
         SPACE 2
         ORG   NKSEDESC
         SPACE 1
*** LAYOUT FOR VOLUME ELEMENT *** 3 VOLUME_ELEMENT DEFINED FILE_ELEMENT,
NKSEVSN  DS    CL6                 4 VOLUME_SERIAL_NUMBER,
NKSEVDT  DS    XL2                 4 VOLUME_DEVICE_TYPE_CODE,
NKSEVTYP DS    XL1                 4 VOLUME_DEVICE_TYPE SET
NKSEVTDI EQU   X'01'                 (DEVICE_TYPE_DISK = 1,
NKSEVTTA EQU   X'02'                  DEVICE_TYPE_TAPE = 2),
NKSEVRES DS    XL1                 4 VOLUME_RESERVATION_TYPE SET
NKSEVRSH EQU   X'00'                 (SHAREABLE = 0,
NKSEVREX EQU   X'01'                  EXCLUSIVE = 1),
NKSEVACC DS    XL1                 4 VOLUME_ACCESS_IF_TAPE SET
NKSEVARD EQU   X'00'                 (READ  = 0,
NKSEVAWR EQU   X'01'                  WRITE = 1),
*
NKSEVLEN EQU   *-NKSETYP          LENGTH_OF_ONE_VOLUME_ELEMENT
         SPACE 2
```

```
         ORG   NKSEDESC
         SPACE 1
*** LAYOUT FOR DEVICE ELEMENT *** 3 DEVICE_ELEMENT DEFINED FILE_ELEMENT,
NKSEDTC  DS    XL2                 4 DEVICE_TYPE_CODE,
NKSEDEV# DS    H                   4 NUMBER_OF_DEVICES,
*
NKSEDLEN EQU   *-NKSETYP          LENGTH_OF_ONE_DEVICE_ELEMENT
         SPACE 2
         ORG   NKSEDESC
         SPACE 1
*** LAYOUT FOR UNIT ELEMENT ***   3 UNIT_ELEMENT DEFINED FILE_ELEMENT,
NKSEUMN  DS    CL4                 4 UNIT_MNEMONIC,                MNE
*
NKSEULEN EQU   *-NKSETYP          LENGTH_OF_ONE_UNIT_ELEMENT
         SPACE 2
         ORG   NKSEDESC                                            102
         SPACE 1                                                   102
*** LAYOUT FOR DEVICE_AT_LOCATION ELEMENT ***                      102
NKSEDTCL DS    XL2                4 TYPE-CODE OF DEVICE_AT_LOCATION 102
NKSEDE#L DS    H                  4 NUMBER OF DEVICE_AT_LOCATION    102
NKSEDLOC DS    CL8                4 DEVICE-LOCATION                 102
*                                                                   102
NKSELLEN EQU   *-NKSETYP                                           102
         SPACE 2                                                   102
         ORG
```

The transferred list of members (elements) does not correspond directly to the parameters specified in the SECURE-RESOURCE-ALLOCATION command.

– In addition to the explicitly specified DISK/TAPE requests, the VOLUME_ELEMENT sublist also contains the implicitly required volumes (supplemented from the catalog entries) specified by means of FILE.

– The DEVICE_AT_LOCATION sublist is a purely additive, more detailed breakdown of the DEVICE sublist. It contains the device requests for which a location allocation exists. The allocation can have been made by:
  – the user specification (LOCATION=<loc>)
  – the default location with *USER-DEFAULT for MAREN
  – a location determined in MAREN-EXIT6 in conjunction with the specification LOCATION=*BY-FILE

LOGON exit (030)

If a LOGON exit routine exists, it is called in the following cases:

– In interactive mode, before task initialization if the user has input an invalid ID, account number or LOGON password. The exit routine can, for example, determine from which terminal the LOGON command was issued.

– After processing a batch or interactive LOGON; the exit routine can, for example, cause the task to be cancelled or system administration procedures to be called.

– In the case of an ENTER-JOB or ENTER-PROCEDURE command from a user (not from an operator) if an incorrect user ID, account number or password was specified. The exit routine can, for example, determine at which terminal the ENTER-JOB or ENTER-PROCEDURE command was issued in interactive mode.

– After successful processing of an ENTER-JOB or ENTER-PROCEDURE command; the exit routine can, for example, compare the caller data with the ENTER data and, if necessary, reject the ENTER job.

The following information is passed to the exit routine:

```
R1    = A(EX030 parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

The return code for the calling system component is transferred by way of the parameter area.

A LOGON or ENTER-JOB command that has already been accepted by the operating system can still be rejected by the exit routine with return code X'04'. The return code should be transferred using the form MF=M of the EX030 macro:

```
EX030    MF=M,PARAM=...,RC=*ACCEPT/*REJECT
```

The user receives the following message:

```
%  JMS0152 SYSTEM ACCESS REQUEST REJECTED BY EXIT ROUTINE
```

Notes

– The contents of registers 12, 13 and 14 must not be destroyed by the exit routine.

– If a SET-LOGON-PARAMETERS, ENTER-PROCEDURE or ENTER-JOB command is issued with invalid user data (JMSXLEJS=JMSXLEIU, JMSXLEIP, JMSXLEIA), the return code has no effect. The command is then rejected with the appropriate error message.

– In the case of ENTER jobs, the LOGON processing that leads to the exit routine being called is performed after the spoolin phase. This means that files generated with /DATA have already been created when the exit routine rejects the LOGON command.

– It is recommended not to conduct dialogs in LOGON exit routines.

DSECT

```
         EXO3O MF=D
         MFTST MF=D,PREFIX=J,MACID=MSX,ALIGN=F,                          C
               DMACID=MSX,SUPPORT=(E,D,C,M,L),DNAME=MSXLE
JMSXLE   DSECT ,
               *,##### PREFIX=J, MACID=MSX #####
*   exit parameterarea description
JMSXLER0 DS    XL6                      reserved
JMSXLERC DS    H                        (main) return code
*   decision of the exit routine
JMSXLEAC EQU   0                        accept job
JMSXLERJ EQU   1                        reject job
*
JMSXLEJP DS    A                        pointer to JTBP
JMSXLEJS DS    FL1                      job state
*   descibes the acceptance state of the job at the time the exit is
*   called
JMSXLEJA EQU   0                        job accepted
JMSXLEIU EQU   4                        job rejected (invalid user
*                                       id)
JMSXLEIP EQU   8                        job rejected (invalid
*                                       password)
JMSXLEIA EQU   12                       job rejected (invalid account
*
JMSXLECL DS    FL1                      instance calling the exit
*   descibes which instance calls the exit
JMSXLEDL EQU   0                        dialog logon
JMSXLEBL EQU   4                        batch logon
JMSXLEEV EQU   8                        enter validation
JMSXLERL EQU   12                       rlogin
*
JMSXLER2 DS    XL2                      reserved
JMSX#    EQU   *-JMSXLER0
```

The JMSXLEJP field contains the address of the JTBP of a job which is about to be defined or started. Changes to this data structure are not permitted. A DSECT of the JTBP can be obtained by calling the $JTBP macro.

The JTBP is only complete when the exit routine is activated during the initialization of a batch task which executes a batch job (JMSXLECL=JMSXLEBL).

The TSN in the JTBP is not yet defined for an accepted ENTER job (JMSXLECL=JMSXLEEV).
The LOGON password is only supplied if the exit routine is activated after validation of the access rights has proved negative (JMSXLEJS=JMSXLEIU,JMSXLEIP,JMSXLEIA).

If the exit routine is activated during the initialization of an accepted dialog or batch task (JMSXLEJS=JMSXLEJA, JMSXLECL=JMSXLEDL, JMSXLEBL), essential data on the initializing task can be obtained by running the JOBINFO macro (see the “Executive Macros” manual [10]).

Job parameter exit (032)

This exit enables the processing of job parameters; these can be defined by systems support during the definition of job classes (cf. the DEFINE-JOB-CLASS statement in the JMU utility routine in the “Utility Routines” manual [4]).

A job parameter is a string of up to 127 characters with freely selectable syntax that can be specified as an operand in an ENTER-PROCEDURE, ENTER-JOB, MODIFY-JOB or SET-LOGON-PARAMETERS command.

The exit routine can only be called in the case of an accepted SET-LOGON-PARAMETERS command in interactive mode or for the ENTER-JOB, ENTER-PROCEDURE and MODIFY-JOB commands. In the case of an ENTER-JOB, ENTER-PROCEDURE or MODIFY-JOB command issued to batch tasks which have not been started, job parameters are also passed to the P1 scheduler, which can also evaluate them.

The following information is passed to the exit routine:

```
R1    = A(EX032 parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

The return code should be transferred to the calling system component using the form MF=M of the EX032 macro.

Note

The job parameter exit is called before any LOGON exit routine. At this point the information available in the system tables is still not complete (e.g. TSN in JTBP).

DSECT

```
         EXO32 MF=D
         MFTST MF=D,PREFIX=J,MACID=MSX,ALIGN=F,                          C
               DMACID=MSX,SUPPORT=(E,D,C,M,L),DNAME=MSXJE
JMSXJE   DSECT ,
               *,##### PREFIX=J, MACID=MSX #####
*   parameterarea description
JMSXJERO DS    XL6                      reserved
JMSXJERC DS    H                        (main) return code
*   decision of the exit routine
JMSXJEAC EQU   0                        accept job
JMSXJERJ EQU   1                        reject job
*
JMSXJEJ  DS    0XL128
JMSXJEJP DS    CL128                    job parameter data
         ORG   JMSXJEJ
*
JMSXJEPD DS    0XL128
JMSXJEPL DS    X                        length of job parameters
JMSXJEPS DS    CL127                    job parameters
*
         ORG   JMSXJEJ+128
JMSXJEJC DS    0XL128
JMSXJECP DS    CL128                    job class parameter data
         ORG   JMSXJEJC
*
JMSXJECD DS    0XL128
JMSXJECL DS    X                        length of job class
*                                       parameters
JMSXJECS DS    CL127                    job class parameters
*
         ORG   JMSXJEJC+128
JMSX#    EQU   *-JMSXJERO
```

JV exit (033)

This exit routine is called when a special job variable that is not supported by the system is accessed. The access can be made either via the SHOW-JV and MODIFY-JV commands or using the GETJV and SETJV macros. Values for the special job variable which are destined for the person using the command can be passed in this exit routine. This makes it possible to implement separate special job variables.
The address of parameter area EX033 is passed in register 1. The JVSXJV field of the parameter area contains the name of the special job variable requested by the user.

The following information is transferred to the exit routine:

```
R1    = A(EX033 parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

Return code

The return code for the calling system component can be transferred in the parameter area. The code can be entered in the JVSXLEN field (1-256) and the contents of the special job variable can be transferred in the JVSXVAL field. The maincode in the standard header must be set to X'0000' and subcode1 must be set to X'00'.

If no exit routine has been specified, the maincode is not equal to X'0000' and/or the subcode is not equal to X'00', the following message is displayed:

```
%  JVS0472 NAME OF SPECIAL JOB VARIABLE NOT PERMITTED. CORRECT COMMAND
```

DSECT

A DSECT for the parameter area (address in register 1) can be generated with the EX033 macro.

```
         EX033 MF=D
         MFTST MF=D,PREFIX=J,MACID=VSX,ALIGN=F,                          C
               DMACID=VSX,SUPPORT=(D,C,M,L),DNAME=VSXPL
JVSXPL   DSECT ,
               *,##### PREFIX=J, MACID=VSX #####
JVSXMIN  EQU   1                          MIN. LENGTH
*
JVSXMAX  EQU   256                        MAX. LENGTH
*
*   end parameterarea
JVSXHDR  FHDR  MF=(C,JVSX),EQUATES=NO                    STANDARDHEADER
JVSXHDR  DS    0A
JVSXFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
JVSXIFID DS    0A              0   INTERFACE IDENTIFIER
JVSXFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
JVSXFCT  DS    AL1             2   FUNCTION NUMBER
JVSXFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
JVSXRET  DS    0A              4   GENERAL RETURN CODE
JVSXSRET DS    0AL2            4   SUB RETURN CODE
JVSXSR2  DS    AL1             4   SUB RETURN CODE 2
JVSXSR1  DS    AL1             5   SUB RETURN CODE 1
JVSXMRET DS    0AL2            6   MAIN RETURN CODE
JVSXMR2  DS    AL1             6   MAIN RETURN CODE 2
JVSXMR1  DS    AL1             7   MAIN RETURN CODE 1
JVSXFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
JVSXSUCC EQU   0                          NO ERROR DETECTED
JVSXREJE EQU   1                          INVALID SPECIAL JV
*
JVSXJV   DS    CL54                       NAME OF THE SPECIAL JV
JVSXLEN  DS    H                          RETURNED LENGTH OF JV-VALUE
JVSXUNU  DS    XL2                        UNUSED
JVSXVAL  DS    CL256                      RETURNED VALUE OF THE SPECIAL
*                                         JV
JVSXUNU1 DS    XL2                        UNUSED
JVSX#    EQU   *-JVSXHDR
```

BCAM exit (036)

This exit permits a check to be kept on requests for connections to system applications (e.g. $DIALOG). The exit routine is called when BCAM recognizes that a local or remote partner wishes to set up a connection to a system application.

The following information is passed to the exit routine:

```
R1    = A($SYSX parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

The parameter area has the following format:

fun	typ1	id1	typ2	id2
0	1	2		

fun = Function of the exit routine (field contents: X'01').

typ = Function type; initially only one type exists, with the code X'01'.
X'01' means monitoring the connections to system applications.

id = Application ID.
id1 is the called application, id2 the calling application.

The application name/processor name are supplied, resulting in the following structure:

01	01	called appl	own proc	01	calling appl	partner proc
00	01	02	0A	12	13	1B 22

called appl: Name of the system application called.

own proc: Name of the local processor.

calling appl: Name of the calling application.

partner proc: Name of the partner processor containing the calling application.

Warning
Only a comparison of names should be performed in the exit routine.
Before the names (IDs) are accessed, the type must be checked because this can change the format (in particular the length) of the ID.
An errored exit routine can cause the communication system (DCM) to crash.

Return code for the calling system component in the $SYSX parameter area.

OPEN exits (039/040/041)

With the aid of three exit points, it is possible to control and monitor use of the “open file” function, regardless of whether the OPEN occurs in a user program, a utility routine (e.g. LMS, EDT) or during the processing of a system command (START-PROGRAM).

– Security exit (039): systems support can reject the OPEN call before it is executed.

– Request exit (041): systems support can reject the OPEN call before it is executed. The file to be opened is defined either from the task file table or from the FCB.

– Return exit (040): the file is already open, and the exit routine can neither prevent nor rescind the opening. All the functions of the return exit can also be performed by the security exit.

The following information is transferred to the exit routine:

R1 = A($DOCSYS parameter area)
R12 = A(TPR program manager)
R13 = A(save area of calling component)
R14 = A(indirect return)
R15 = A(exit routine)

Notes

– Registers 12, 13 and 14 must not be destroyed by the exit routine.

– Depending on the value of the AUDIT indicator, the 039/040 exits can carry out certain monitoring activities in the catalog entry (see “File access monitoring” on page 12).

– If either of exits 039/040 is activated, the BS2000/OSD Data Management System (DMS) has already opened the file. The catalog entry for the opened file is still locked in exits 039 and 041. If a programming error causes the exit routine to be hung up or enter a loop without terminating abnormally, the file remains locked until the end of the session. (If it was terminated abnormally, this would lead to abnormal task termination and the file would be released.)

– In exit 040 the catalog entry for the opened file has already been released.

– The exit routine is allowed to reopen files on its own authority. Any problems resulting from recursive call nesting must be dealt with by the exit routine itself.

– When working under the TSOS user ID it is possible, in the case of tape files, to use these exits and corresponding exit routines to monitor and keep a record of the use of tapes. This enables you to implement your own tape management facility.

– If the attempt to open the file is rejected, the user receives either message DMS0D10 on SYSOUT or error code X’0D10’ in the ID1ECB field of the FCB:

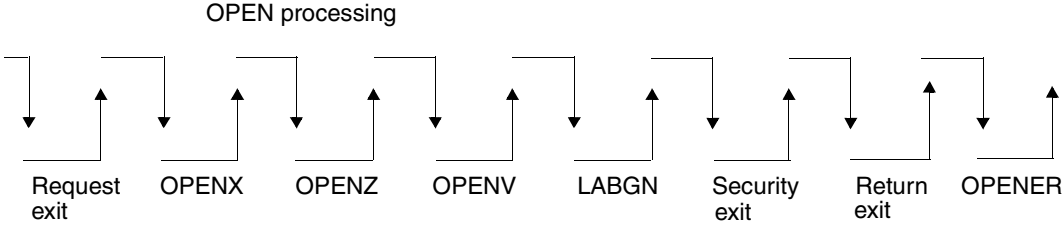
```
%  DMSOD10 SYSTEM ADMINISTRATOR HAS RESTRICTED USE OF OPEN MACRO.
The activated system exit routine has rejected the OPEN command or macro.
The program is continued with the EXLST exit OPENER, if defined, otherwise
it is aborted.
```

– If there is an error in the exit base mechanism, message DMS0D15 is displayed on the console and then processing continues.

```
%  DMSOD15 CALLING BASE MECHANISM OF SYSTEM EXIT '(&00)' RESULTS IN ERROR
CODE '(&01)'
(&00): exit number.
(&01): error code of $SYSX interface
```

Exit routines and EXLST exits

The exit routines described above and the EXLST exits that can be connected during OPEN processing are executed in the order shown in the diagram below (if they are present):



Parameter area

The layout of the exit parameter area can be generated with the $DOCSYS macro.

```
         $DOCSYS MF=D
         MFTST MF=D,PREFIX=D,MACID=OCS,ALIGN=F,                          C
               DMACID=OCS,SUPPORT=(D,C,M,L),DNAME=OCS_MDL
DOCS_MDL DSECT ,
               *,##### PREFIX=D, MACID=OCS #####
*   parameterarea description
DOCSHDR  FHDR  MF=(C,DOCS),EQUATES=NO                    Standardheader
DOCSHDR  DS    0A
DOCSFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
DOCSIFID DS    0A              0   INTERFACE IDENTIFIER
DOCSFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
DOCSFCT  DS    AL1             2   FUNCTION NUMBER
DOCSFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
DOCSRET  DS    0A              4   GENERAL RETURN CODE
DOCSSRET DS    0AL2            4   SUB RETURN CODE
DOCSSR2  DS    AL1             4   SUB RETURN CODE 2
DOCSSR1  DS    AL1             5   SUB RETURN CODE 1
DOCSMRET DS    0AL2            6   MAIN RETURN CODE
DOCSMR2  DS    AL1             6   MAIN RETURN CODE 2
DOCSMR1  DS    AL1             7   MAIN RETURN CODE 1
DOCSFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
DOCSNOER EQU   0                       no error detected
DOCSRREJ EQU   4                       request rejected
*
DOCSFCB@ DS    A                       address of P1-FCB
DOCS#    EQU   *-DOCSHDR
```

Security exit (039)

The return code for the calling system module is transferred in the standard header of the $DOCSYS parameter area in the DOCSMRET field:

Return code: DOCSNOER = X'0000' = no error

Return code: DOCSRREJ = X'0004' = OPEN is to be rejected

The security exit is not activated if OPEN is rejected due to an invalid file name.

Return exit (040)

The return exit is activated even if OPEN is rejected due to an invalid file name.

The exit routine cannot send a return code to the system module.

Request exit (041)

The return code for the calling system module is transferred in the standard header of the $DOCSYS parameter area in the DOCSMRET field:

Return code: DOCSNOER = X'0000' = no error

Return code: DOCSRREJ = X'0004' = OPEN is to be rejected

CLOSE exits (042/043)

Two exits are offered for the CLOSE (= “close file”) function. For tape files, the corresponding exit routines can be used under the TSOS user ID to monitor and keep a record of the use of tapes, and thus to implement an individual tape file management facility.

– The request exit is activated at the beginning of CLOSE processing. In the case of magnetic tape files the tape is positioned behind the last data block processed. End-of-file labels have not yet been processed.
– The return exit routine is called at the end of CLOSE processing. In the case of magnetic tape files the tape is positioned between the two tape marks which serve as delimiters at the end of a file; for files with standard labels, this position is behind the EOF labels.

In both cases the address of the exit parameter list is transferred to the exit routine. It is therefore possible to access further information from the P1 FCB, P2 FCB or the CVT (Current Volume Table).

The following information is passed to the exit routine.

```
R1    = A($DOCSYS parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

Notes

– Registers 12, 13 and 14 must not be destroyed by the exit routine.

– If the return code in the parameter area of the exit base mechanism is not equal to 0, message DMS0D15 is displayed on the console and then processing continues.

```
%  DMS0D15 CALLING BASE MECHANISM OF SYSTEM EXIT '(&00)' RESULTS IN ERROR
CODE '(&01)'
(&00): exit number.
(&01): error code of $SYSX interface
```

Sources of information

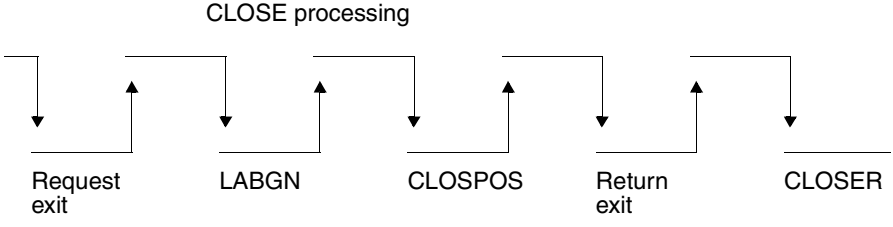
The following table contains notes on accessing information which can be processed effectively by the exit routines:

Information	DSECT		Name	is in	Note
Exit parameter list	$DOCSYS	A		Register 1	
P1-FCB	IDFCB	A	P1FCB-ADDRESS	Exit parameter list	
P2-FCB	IDFC2	A	D1P2LNK	P1-FCB	
CVT	DMACVT	A	D2CVTAD	P2-FCB	
VSN	DMADR	I	DRVVSN	TFT	with RDTFT macro
Device access right	DMACVT	I	DCVRID	CVT	is not available with the CLOSE return exit
Labels	DTVOL DTHT1 DTHT2 DTHT3	A A	D2LGINF D2LGINF	P2-FCB P2-FCB	return exit request exit
Total number of block counters for file		I	D2THPGN	P2-FCB	only for SAM and CLOSE return exit
Block counters for tape		I I	DIHPGNR D1BLK# / DIBLK#	P2-FCB P1-FCB / P2-FCB	BLKSIZE=STD BLKSIZE (STD,n) — for SAM only

I = Information is contents of “Name” field
A= Address of information in “Name” field

Exit routines and EXLST exits

The exit routines described above and the EXLST exits that can be connected during CLOSE processing are executed in the order shown in the diagram below (if they are present):



Return exit (042)

The exit routine cannot send return codes to the system module.

Request exit (043)

The exit routine cannot send return codes to the system module.

In the case of output files on tape, the file labels VOL1, HDR1, HDR2 and HDR3 written with OPEN can be accessed via the address in the D2LGINF field in the P2 FCB. The length of the addressed area is 4 x 80 = 320 bytes.

When the request exit routine is called, no value has been specified for the block counter.

Parameter area

The layout of the exit parameter area can be generated with the $DOCSYS macro.

```
         $DOCSYS MF=D
         MFTST MF=D,PREFIX=D,MACID=OCS,ALIGN=F,                          C
               DMACID=OCS,SUPPORT=(D,C,M,L),DNAME=OCS_MDL
DOCS_MDL DSECT ,
               *,##### PREFIX=D, MACID=OCS #####
*   parameterarea description
DOCSHDR  FHDR  MF=(C,DOCS),EQUATES=NO                    Standardheader
DOCSHDR  DS    0A
DOCSFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
DOCSIFID DS    0A              0   INTERFACE IDENTIFIER
DOCSFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
DOCSFCT  DS    AL1             2   FUNCTION NUMBER
DOCSFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
DOCSRET  DS    0A              4   GENERAL RETURN CODE
DOCSSRET DS    0AL2            4   SUB RETURN CODE
DOCSSR2  DS    AL1             4   SUB RETURN CODE 2
DOCSSR1  DS    AL1             5   SUB RETURN CODE 1
DOCSMRET DS    0AL2            6   MAIN RETURN CODE
DOCSMR2  DS    AL1             6   MAIN RETURN CODE 2
DOCSMR1  DS    AL1             7   MAIN RETURN CODE 1
DOCSFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
DOCSNOER EQU   0                       no error detected
DOCSRREJ EQU   4                       request rejected
*
DOCSFCB@ DS    A                       address of P1-FCB
DOCS#    EQU   *-DOCSHDR
```

Volume swap exits (044/045)

In addition to the CLOSE exits for magnetic tape files (see above: 042/043) two exits are offered for intervention during volume swapping:

– the request exit routine is activated at the beginning of every tape volume swap, i.e. during the end processing of the tape to be closed
– the return exit routine is called after the start processing of the new magnetic tape volume.

As for the CLOSE exits for tape files, the address of the exit parameter area, which contains the address of the P1-FCB, in passed in both cases.

For tape files, the corresponding exit routines can be used under the TSOS user ID to monitor and keep a record of the use of tapes, and thus to implement an individual tape file management facility.

The following information is transferred to the exit routine:

```
R1    = A($DOCSYS parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

Notes

– Registers 12, 13 and 14 must not be destroyed by the exit routine.

– If there is an error in the exit base mechanism, message DMS0D15 is displayed on the console and then processing continues.

```
%  DMSOD15 CALLING BASE MECHANISM OF SYSTEM EXIT '(&OO)' RESULTS IN ERROR
CODE '(&O1)'
(&OO): exit number.
(&O1): error code of $SYSX interface
```

Parameter area

The layout of the exit parameter area can be generated with the $DOCSYS macro.

```
         $DOCSYS MF=D
         MFTST MF=D,PREFIX=D,MACID=OCS,ALIGN=F,                          C
               DMACID=OCS,SUPPORT=(D,C,M,L),DNAME=OCS_MDL
DOCS_MDL DSECT ,
               *,##### PREFIX=D, MACID=OCS #####
*   parameterarea description
DOCSHDR  FHDR  MF=(C,DOCS),EQUATES=NO                    Standardheader
DOCSHDR  DS    0A
DOCSFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
DOCSIFID DS    0A              0   INTERFACE IDENTIFIER
DOCSFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
DOCSFCT  DS    AL1             2   FUNCTION NUMBER
DOCSFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
DOCSRET  DS    0A              4   GENERAL RETURN CODE
DOCSSRET DS    0AL2            4   SUB RETURN CODE
DOCSSR2  DS    AL1             4   SUB RETURN CODE 2
DOCSSR1  DS    AL1             5   SUB RETURN CODE 1
DOCSMRET DS    0AL2            6   MAIN RETURN CODE
DOCSMR2  DS    AL1             6   MAIN RETURN CODE 2
DOCSMR1  DS    AL1             7   MAIN RETURN CODE 1
DOCSFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
DOCSNOER EQU   0                       no error detected
DOCSRREJ EQU   4                       request rejected
*
DOCSFCB@ DS    A                       address of P1-FCB
DOCS#    EQU   *-DOCSHDR
```

Return exit (044)

The exit routine cannot send a return code to the system module. In the case of output files, the address of a 320-byte area containing the VOL1, HDR1, HDR2 and HDR3 labels is contained in D2LGINF in the P2-FCB.

Return exit (045)

The return code is passed in the standard header of the parameter area in the DOCSMRET field.

Return code: DOCSNOER = X'0000' = no error

Return code: DOCSRREJ = X'0004' = OPEN is to be rejected

If the attempt to carry out a volume swap is rejected, the user receives message DMS0E10 on SYSOUT or error code X’0E10’ in the ID1ECB field of the FCB:

```
DMS0E10  SYSTEM ADMINISTRATION RESTRICTS THE USE OF VOLUME SWITCHING
```

In the case of output files, the address of a 320-byte area containing the VOL1, EOF1, EOF2 and EOF3 labels is contained in D2LGINF in the P2-FCB.

FILE exits (050/051)

Use of the following commands can be monitored at two exit points with the aid of an exit routine:
– CREATE-FILE
– MODIFY-FILE-ATTRIBUTES
– CREATE-FILE-GENERATION
– MODIFY-FILE-GENERATION-SUPPORT
– ADD-FILE-LINK
– CREATE-TAPE-SET
– IMPORT-FILE (for tape files)

Furthermore, it is also possible to monitor the FILE macro.

The request exit routine can reject or modify a FILE request.

The return exit routine can initiate the appropriate logging or recording actions after the execution of a FILE request.

Return exit (050)

The following information is passed to the exit routine:

```
R1    = A(EXO50 parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

The exit routine cannot send a return code to the system module. The return from the return exit routine is followed by the return from FILE processing. The exit routine must set the value X'0000' or X'0004' for the maincode in the standard header of the parameter area. If the maincode contains any other value, the following message appears on the console:

```
%  DMSO6B9 CALLING SYSTEM EXIT (&OO) RESULTS IN ERROR CODE (&O1)
```

Request exit (051)

The register settings are the same as for the return exit.

The return code for the calling system module is sent in the maincode of the standard header of the exit parameter area. The following values are possible:

X'0000' FILE processing is performed with the FILE parameter area, which may have been modified by the exit routine.

X'0004' The FILE request is rejected with error code X'0515'.
For commands, the following message is sent to SYSOUT:
`DMSO515 USE OF SPECIFIED COMMAND OR 'FILE' MACRO RESTRICTED BY SYSTEM ADMINISTRATOR. COMMAND OR MACRO REJECTED.`

other Same as for X'0004'; in addition the following message is displayed on the console:
`DMSO6B9 CALLING SYSTEM EXIT (&OO) RESULTS IN ERROR CODE (&O1)`

Note

– Registers 12, 13 and 14 must not be destroyed by the exit routine.

DSECT

```
         EX050 MF=D
         MFTST MF=D,PREFIX=D,MACID=MAX,ALIGN=F,                          C
               DMACID=MAX,SUPPORT=(D,C,L),DNAME=MAXMDL
DMAXMDL  DSECT ,
               *,##### PREFIX=D, MACID=MAX #####
*   Parameterarea
DMAXHDR  FHDR  MF=(C,DMAX),EQUATES=NO                  Standard-Header
DMAXHDR  DS    0A
DMAXFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
DMAXIFID DS    0A              0   INTERFACE IDENTIFIER
DMAXFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
DMAXFCT  DS    AL1             2   FUNCTION NUMBER
DMAXFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
DMAXRET  DS    0A              4   GENERAL RETURN CODE
DMAXSRET DS    0AL2            4   SUB RETURN CODE
DMAXSR2  DS    AL1             4   SUB RETURN CODE 2
DMAXSR1  DS    AL1             5   SUB RETURN CODE 1
DMAXMRET DS    0AL2            6   MAIN RETURN CODE
DMAXMR2  DS    AL1             6   MAIN RETURN CODE 2
DMAXMR1  DS    AL1             7   MAIN RETURN CODE 1
DMAXFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
DMAXFRC  DS    F                       FILE-Returncode
DMAXFPL  DS    A                       Addr der FILE-PL
DMAX#    EQU   *-DMAXHDR
```

In the case of the return exit, the MAXFRC field contains the return code of FILE processing.

FILE parameter list

A DSECT for the FILE parameter area is generated by means of the FILE macro:

```
[name]   FILE  MF=D[,PREFIX=prefix/*],VERSION=3
```

The file parameter area must be generated with the format valid as of BS2000/OSD-BC V3.0.
The MAXFPL field of the EX050 DSECT contains the address of the file parameter area.

ATTACH library exit (054)

This exit point enables systems support to specify which program libraries are to be monitored. The exit routine is activated after every ATTACH library operation (open program library). With the aid of the system exit systems support determines whether the member accesses to the currently monitored library are to be checked.

The following information is passed to the exit routine:

```
R1    = A(exit parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

(cf. chapter “System exits for program libraries” on page 149)

Information in the parameter area:

– library name
– link name of the library
– PLAM return code 1
– PLAM return code 2

If the library was opened correctly, the PLAM return codes contain the value 0.

Return code:

```
R15 =    X'000000RR'

         RR =   Return code of exit routine
                00  library is to be monitored
                04  library is not to be monitored
```

DSECT

The DSECT for the parameter area can be generated by means of the EX054 macro:

```
name EX054 D[,[prefix],[ALL]]
```

```
         EX054 D
IEX054   DSECT
         DS    0F
*****************************************************
* ATTACH  PARAMETER LIST FOR PLAM RZ ROUTINES       *
*****************************************************
PMRC     DC    F'0'                    PLAM RETURN CODE
PMRC1    DC    F'0'                    PLAM SECONDARY RETURN CODE
PMPLV    DC    CL8' '                  VERSION OF PLAM: VZZ.ZBZZ
PMLIB    DC    A(0)                    A(LIBRARY NAME):   CL54
PMLINK   DC    A(0)                    A(LIBRARY NAME LINK):   CL8
         DC    CL1' '
         DC    CL1' '
         DC    CL1' '
         DC    CL1' '
         DC    A(0)
         DC    A(0)
         DC    A(0)
         DC    A(0)
         DC    A(0)
         DC    A(0)
         DC    A(0)
         DC    A(0)
LEN      EQU   *-PMRC
```

DELETE member exits (055/056)

Systems support can monitor the deletion of program library members by means of two exit points.

– At the request exit systems support can reject the DELETE call before it is executed.

– At the return exit the member has already been deleted. Here systems support receives the full identification of the deleted member.

Request exit (055)

The following information is passed to the exit routines:

```
R1    = A(exit parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

(cf. chapter “System exits for program libraries” on page 149)

Return code:

```
R15 =    X'000000RR'

         RR =   Return code of exit routine
                00  no error
                04  (DELETE member rejected)
```

Return exit (056)

The return exit is identical to the request exit.

DSECT

The DSECT for the parameter area can be generated by means of the EX055 macro:

```
name EX055 D[,[prefix],[ALL]]
```

```
         EX055 D
IEX055   DSECT
         DS    0F
*****************************************************
* DELETE  PARAMETER LIST FOR PLAM RZ ROUTINES       *
*****************************************************
PMRC     DC    F'0'                    PLAM RETURN CODE
PMRC1    DC    F'0'                    PLAM SECONDARY RETURN CODE
PMPLV    DC    CL8' '                  VERSION OF PLAM: VZZ.ZBZZ
PMLIB    DC    A(0)                    A(LIBRARY NAME):   CL54
PMLINK   DC    A(0)                    A(LIBRARY NAME LINK):   CL8
         DC    CL1' '
         DC    CL1' '
         DC    CL1' '
         DC    CL1' '
PMTYP    DC    A(0)                    A(MEMBER TYPE):    CL8
PMNAM    DC    A(0)                    A(MEMBER NAME):    CL64
PMVER    DC    A(0)                    A(MEMBER VERSION): CL24
PMVAR    DC    A(0)                    A(MEMBER VARIANT): F
PMDAT    DC    A(0)                    A(MODIFICATION DATE): CL14: DATE
*                                                             ,CL8:  TIME
         DC    A(0)
         DC    A(0)
         DC    A(0)
LEN      EQU   *-PMRC
```

RENAME member exits (057/058)

Systems support can monitor the renaming of program library members by means of two exit points:

– At the request exit systems support can reject the RENAME call before the function is performed.

– At the return exit the member has already been renamed. Here systems support receives the full identification of the member, i.e. both the original and the new identification.

Request exit (057)

The following information is passed to the exit routines:

```
R1    = A(exit parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

(cf. chapter “System exits for program libraries” on page 149)

Return code:

```
R15 =    X'000000RR'

         RR =   Return code of exit routine
                00  no error
                04  (DELETE member rejected)
```

Return exit (058)

The return exit is identical to the request exit.

DSECT

The DSECT for the parameter area can be generated by means of the EX057 macro:
`name EX057 [D]`

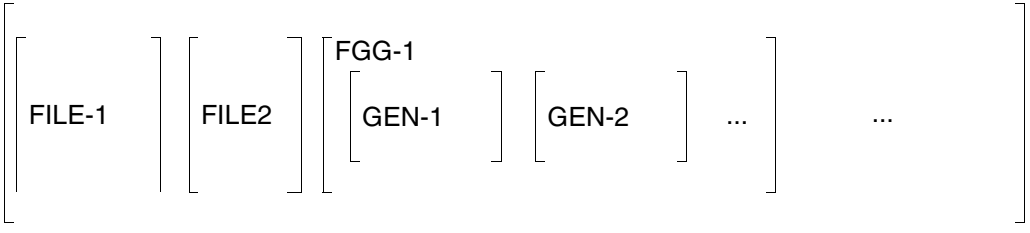
```
         EX057 D
IEX057   DSECT
         DS    0F
*****************************************************
* RENAME  PARAMETER LIST FOR PLAM RZ ROUTINES       *
*****************************************************
PMRC     DC    F'0'                    PLAM RETURN CODE
PMRC1    DC    F'0'                    PLAM SECONDARY RETURN CODE
PMPLV    DC    CL8' '                  VERSION OF PLAM: VZZ.ZBZZ
PMLIB    DC    A(0)                    A(LIBRARY NAME):   CL54
PMLINK   DC    A(0)                    A(LIBRARY NAME LINK):   CL8
         DC    CL1' '
         DC    CL1' '
         DC    CL1' '
         DC    CL1' '
PMTYP    DC    A(0)                    A(MEMBER TYPE):    CL8
PMNAM    DC    A(0)                    A(MEMBER NAME):    CL64
PMVER    DC    A(0)                    A(MEMBER VERSION): CL24
PMVAR    DC    A(0)                    A(MEMBER VARIANT): F
PMDAT    DC    A(0)                    A(MODIFICATION DATE): CL14: DATE
*                                                             ,CL8:  TIME
PMTYP2   DC    A(0)                    A(MEMBER TYPE 2):    CL8
PMNAM2   DC    A(0)                    A(MEMBER NAME 2):    CL64
PMVER2   DC    A(0)                    A(MEMBER VERSION 2): CL24
LEN      EQU   *-PMRC
```

ERASE exits (060/061)

Two exit points are provided for ERASE processing, a request exit and a return exit. They each contain one job.

A job is always one of the commands DELETE-FILE, DELETE-FILE-GROUP, DELETE-FILE-GENERATION, DELETE-SYSTEM-FILE, EXPORT-FILE or the ERASE macro, or any catalog entry that is to be processed by ERASE.

File generation groups, however, are special cases. Here the catalog entries of the generations do not form separate jobs. Rather, the file generation group as a whole is regarded as a job.



– In the case of the request exit the exit routine can check the job and reject it if appropriate. If monitoring is to be performed by the return exit routine, the AUDIT indicator from the catalog entry must be saved.
– The program branches to the return exit once the job has been processed. This exit routine can check the DMS error code and initiate any monitoring action that may be necessary.

In register 1 both exits contain a pointer to a structure that is described by the EX060 macro and stored in the working area of the calling system module.

If systems support wishes to implement monitoring measures in the exit routine that depend on the AUDIT value in the catalog entry, this information must already be provided at the request exit, since after a successful deletion (at the return exit) the catalog entry is normally no longer available.

The field for the file name is 80 bytes long. This field should be cleared in the exit routine before the name is provided, as it might otherwise contain smudge characters. If the field is not cleared, the end criterion for the file name must be requested (X'40').

Return exit (060)

The following information is passed to the exit routine:

```
R1    = A(EX060 exit parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

The return exit serves to indicate the success of an erase job. It is not called if the request exit routine has rejected the ERASE request.

Whenever this exit is called, field &p.RET of the ERASE parameter area contains the return code supplied during erasure or attempted erasure. If no error occurs, it contains the value X’00000000’. The ERASE parameter area is addressed via the MAXEPL field. The MAXFLC field contains the value contained by the corresponding MAXFLC field of the request exit. If MAXFLC is zero, the return exit of a command or macro is involved, and &p.RET contains the generic error code.

The exit routine cannot send a return code to the system module. The return from the return exit routine is followed by the return from ERASE processing. The exit routine must set the value X'70000' or X'0004' for the maincode in the standard header of the parameter area. If the maincode contains any other value, the following message appears on the console:

```
%  DMS06B9 CALLING SYSTEM EXIT (&00) RESULTS IN ERROR CODE (&01)
```

Request exit (061)

The register settings are the same as for the return exit.

Control branches to the request exit exactly once per ERASE job. Before this happens, the EX060 parameter block is supplied with the following values:

MAXEPL points to the ERASE parameter area (see above); MAXFLC contains a counter, which starts at zero with every command or macro request and is incremented by one with every new job.

The return code for the calling system module is sent in the maincode of the standard header of the exit parameter area. The following values are possible:

X'0000' ERASE processing is performed with the ERASE parameter area, which may have been modified by the exit routine.

X'0004' The ERASE request is rejected with error code X'0514'.
For commands, the following message is sent to SYSOUT:
`DMS0514 USE OF SPECIFIED COMMAND OR 'ERASE' MACRO RESTRICTED BY SYSTEM ADMINISTRATOR. COMMAND OR MACRO REJECTED.`

other Same as for X'0004'; in addition the following message is displayed on the console:
`DMS06B9 CALLING SYSTEM EXIT (&00) RESULTS IN ERROR CODE (&01)`

Note

– Registers 12, 13 and 14 must not be destroyed by the exit routine.

DSECT

```
         EXO6O MF=D
         MFTST MF=D,PREFIX=D,MACID=MAX,ALIGN=F,                          C
               DMACID=MAX,SUPPORT=(D,C,L),DNAME=MAXMDL
DMAXMDL  DSECT ,
               *,##### PREFIX=D, MACID=MAX #####
*   Parameterarea
DMAXHDR  FHDR  MF=(C,DMAX),EQUATES=NO                  Standard-Header
DMAXHDR  DS    0A
DMAXFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
DMAXIFID DS    0A              0   INTERFACE IDENTIFIER
DMAXFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
DMAXFCT  DS    AL1             2   FUNCTION NUMBER
DMAXFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
DMAXRET  DS    0A              4   GENERAL RETURN CODE
DMAXSRET DS    0AL2            4   SUB RETURN CODE
DMAXSR2  DS    AL1             4   SUB RETURN CODE 2
DMAXSR1  DS    AL1             5   SUB RETURN CODE 1
DMAXMRET DS    0AL2            6   MAIN RETURN CODE
DMAXMR2  DS    AL1             6   MAIN RETURN CODE 2
DMAXMR1  DS    AL1             7   MAIN RETURN CODE 1
DMAXFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
DMAXFLC  DS    F                       File Counter
DMAXEPL  DS    A                       Addr der ERASE-PL
DMAX#    EQU   *-DMAXHDR
```

The MAXEPL field contains a pointer to a version 3 parameter area. If a version 3 ERASE has been requested, the area is a copy of the original parameter area. If a version 0, version 1, or version 2 ERASE has been requested, the old format is converted to the new format. It is impossible to access the original parameter area.

ERASE parameter list

The DSECT for the ERASE parameter area can be generated as follows:

```
[name]    ERASE MF=D[,PREFIX=prefix/*],VERSION=3
```

The ERASE parameter area must be generated in the format valid as of BS2000/OSD-BC V3.0 (VERSION=3). All exit routines which access the ERASE parameter area have to be adapted to handle this new format.

CATALOG exits (070/071)

Use of the following commands can be monitored at two exit points with an exit routine.
– CREATE-FILE
– CREATE-FILE-GENERATION
– CREATE-FILE-GROUP
– MODIFY-FILE-ATTRIBUTES
– MODIFY-FILE-GENERATION-SUPPORT
– MODIFY-FILE-GROUP-ATTRIBUTES.

In addition, it is possible to monitor the CATAL macro.

The request exit routine can reject or modify a CATALOG request.

The return exit routine can initiate the appropriate logging or recording actions after the execution of a CATALOG request.

Return exit (070)

The following information is passed to the exit routine:

R1 = A(EX070 parameter area)
R12 = A(TPR program manager)
R13 = A(save area of calling component)
R14 = A(indirect return)
R15 = A(exit routine)

The exit routine cannot send a return code to the system module. The return from the return exit routine is followed by the return from CATALOG processing. The exit routine must set the value X'0000' or X'0004' for the maincode in the standard header of the parameter area. If the maincode contains any other value, the following message appears on the console:

```
%  DMS06B9 CALLING SYSTEM EXIT (&00) RESULTS IN ERROR CODE (&01)
```

Request exit (071)

The register settings are the same as for the return exit.

The return code for the calling system module is sent in the maincode of the standard header of the exit parameter area. The following values are possible:

X'0000' CATALOG processing is performed with the CATALOG parameter area, which may have been modified by the exit routine.

X'0004' The CATALOG request is rejected with error code X'0513'.
For commands, the following message is sent to SYSOUT:
`DMS0513 USE OF SPECIFIED COMMAND OR 'CATAL' MACRO RESTRICTED BY SYSTEM ADMINISTRATOR. COMMAND OR MACRO REJECTED.`

other Same as for X'0004'; in addition the following message is displayed on the console:
`DMS06B9 CALLING SYSTEM EXIT (&00) RESULTS IN ERROR CODE (&01)`

Notes

– Registers 12, 13 and 14 must not be destroyed by the exit routine.

– The exit routine can set or modify the AUDIT parameter without the user having any means of influencing or preventing the change.

DSECT

```
         EXO7O MF=D
         MFTST MF=D,PREFIX=D,MACID=MAX,ALIGN=F,                          C
               DMACID=MAX,SUPPORT=(D,C,L),DNAME=MAXMDL
DMAXMDL  DSECT ,
               *,##### PREFIX=D, MACID=MAX #####
*   Parameterarea
DMAXHDR  FHDR  MF=(C,DMAX),EQUATES=NO                  Standard-Header
DMAXHDR  DS    OA
DMAXFHE  DS    OXL8            0   GENERAL PARAMETER AREA HEADER
*
DMAXIFID DS    OA              0   INTERFACE IDENTIFIER
DMAXFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
DMAXFCT  DS    AL1             2   FUNCTION NUMBER
DMAXFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
DMAXRET  DS    OA              4   GENERAL RETURN CODE
DMAXSRET DS    OAL2            4   SUB RETURN CODE
DMAXSR2  DS    AL1             4   SUB RETURN CODE 2
DMAXSR1  DS    AL1             5   SUB RETURN CODE 1
DMAXMRET DS    OAL2            6   MAIN RETURN CODE
DMAXMR2  DS    AL1             6   MAIN RETURN CODE 2
DMAXMR1  DS    AL1             7   MAIN RETURN CODE 1
DMAXFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
DMAXCRC  DS    F                       CATAL-Returncode
DMAXCPL  DS    A                       Addr der CATAL-PL
DMAX#    EQU   *-DMAXHDR
```

In the case of the return exit, the MAXCRC field contains the return code of CATAL processing

CATALOG parameter list

The DSECT for the CATALOG parameter area is generated by means of the CATAL macro:

```
[name]    CATAL MF=D[,PREFIX=prefix/*],VERSION=3
```

The MAXCPL field of the exit parameter area contains a pointer to the CATALOG parameter area. The CATALOG parameter area must be generated in the format valid as of BS2000/OSD-BC V3.0 (VERSION=3).

DCAM exit (075)

The DCAM exit permits the user to expand and modify DCAM functions and to add special functions. Examples of such functions are:

– data protection facilities over and above those in TRANSDATA by means of more extensive authorization checks; e.g. make the opening of applications dependent on the user ID, permit a connection to be set up for an application only with defined partners or processors

– initiation of additional actions (starting jobs, issuing messages).

The user-specific exit routine is invoked for opening and closing applications and for connection setup and cleardown. When opening applications and setting up connections, the exit routine can accept, reject or, if necessary, modify the call. The calls on closing serve only to provide the exit routine with information.

All information refers to DCAM as of V13.0A.

Exit cases (events)

There are currently 12 exit cases for the DCAM exit routine. A distinction is made between exit main cases and exit subcases, which are identified by means of a sequence of 3 alphabetic characters each. The 6-digit combination of the main case and subcase identifiers uniquely defines the exit case.

The expansion of exit main case and exit subcase in future DCAM versions must be taken into account.

The main cases APP and CON, together with their subcases are compared on the following pages.

Main case APP

This refers to cases concerned with opening and closing applications.

Exit subcase	Exit routine is called	Exit routine functions
APPOPN	on all YOPEN calls, after checking the user specifications prior to processing the call.	- additional checks e.g. applications may be opened by certain users only; load tables dynamically. - accept YOPEN call - reject YOPEN call - modify YOPEN parameters - initiate additional actions e.g. start jobs; output messages etc.
APPCLS	on all YCLOSE calls, prior to clearing down the DCAM data structure.	- initiate additional actions; e.g. output messages, update exit-specific tables etc.
APPSHU	on internal closure of an application after the operator has entered the BCEND, BCLOSE or SHUTDOWN command *)	
APPRTM	on internal closure of an application when a program or task is terminated	initiate a substitute task
APPFCL	on unexpected closure of an application due to DCAM failure.	

*) The exit is only called on closing and not in the event of any previous warning. The exit is not called if TIAM termination precedes DCAM termination, i.e. if the DCAM program was started as an interactive task. In such a case, task termination will be initiated by TIAM, i.e. the APPTRM exit is called.

Main case CON

This refers to cases concerned with connection setup and cleardown. The exit routine is not called, however, in the case of an implicit cleardown of all existing connections when closing an application.

A distinction is made between the following subcases:

– Request connection setup (YOPNCON ACQUIRE):
  – on YOPNCON ACQUIRE call by user
  – after successful connection setup
– Accept connection request
  – on receipt of a connection setup request
  – after acceptance of the request (YOPNCON ACCEPT)
– In the case of explicit connection cleardown or after a DCAM error.
  – connection cleardown by user
  – connection cleardown forced by breakdown
  – connection cleardown by partner

Exit subcase	Exit routine is called	Exit routine functions
CONACQ	on sending a connection setup request (YOPNCON ACQUIRE), after checking user specifications, prior to call processing (i.e. prior to the BCAM call REQCON)	- additional checks e.g. does connection message (LOGON MESSAGE) conform to conventions? - accept call - reject call - modify YOPNCON parameters
CONACC	on every acceptance of a connection request (YOPNCON ACCEPT), after verifying user specifications, prior to call processing	- additional checks - accept request - reject request - modify YOPNCON parameters
CONCPL	after a successful connection setup, i.e. if a YOPNCON ACQUIRE request has been accepted by the partner	
CONREQ	on receipt of a connection request, after checking specifications, prior to processing the connection request.	- additional checks - accept connection request - reject connection request

Exit subcase	Exit routine is called	Exit routine functions
CONCLS	on every connection cleardown by the user (YCLSCON), prior to release of the DCAM data structure	
CONFCL	on each forced connection cleardown due to an NEA logging error or an acknowledgment overflow	
CONBAD	on each connection cleardown initiated by the partner.	

The following information is passed to the exit routine:

```
R1    = A(YDDEXPL) = DCAM exit parameter area
R12   = A(TPR program manager)
R13   = A(save area)
R14   = A(indirect return)
R15   = A(exit routine)
```

The return code for the calling system component is sent via the parameter area in the YDDXRETC field:

```
YDDXRETC  X'00FDBKRC'

          FDBK = Exit routine return code for the DCAM user.

          RC     Exit routine return code for the calling DCAM component.
          =      It can have the following values:
                 00  accept call
                 04  reject call
                 08  modify call
```

The FDBK field must be loaded when RC=04 is set. This specification is transferred to the DCAM user in the FDBK1/ FDBK2 field (see the “DCAM Macros” manual [1]). FDBK can either be taken from the area reserved for the exit routine ’CC00’-’CCFF’ or it can be one of the valid DCAM return codes (see the “System Messages” [16]). FDBK must not be ’0000’. If RC=08 is set, the exit routine stores the data for modification in the YDDEXPL parameter area. Due attention should be paid to the notes on programming exit routines.

With the exit events APPOPN, CONACQ and CONACC, the FDBK field is only passed to the user (equivalent to YOPEN, YOPNCON ACQ and YOPNCON ACC) but is not sent to the partner.

If the system exit of the partner rejects a connection request, the YOPNCON ACQUIRE is terminated with X'0C40' instead of the FDBK specified in that system exit.

DSECT

```
         YDDEXPL D
YDDEXPL  DSECT
*********************************************************************
*                                                                   *
*    DCAM  EXIT PARAMETERLISTE                                      *
*                                                                   *
*********************************************************************
*
* EINGABEPARAMETER
*
YDDXDPAR DS    A             A(DARPPAR)
*                            DCAM INTERNER ARBEITSBEREICH
YDDXMAIN DS    X             EXIT-MAIN-CASE
YDDXAPPL EQU   1               APPL
YDDXCONN EQU   2               CONN
YDDXSUB DS    X             EXIT-SUB-CASE
YDDXAOPN EQU   1               APPOPN
YDDXACLS EQU   2               APPCLS
YDDXASHU EQU   3               APPSHU
YDDXATRM EQU   4               APPTRM
YDDXAFCL EQU   5               APPFCL
YDDXCACQ EQU   6               CONACQ
YDDXCACC EQU   7               CONACC
YDDXCCPL EQU   8               CONCPL
YDDXCREQ EQU   9               CONREQ
YDDXCCLS EQU   10              CONCLS
YDDXCFCL EQU   11              CONFCL
YDDXCBAD EQU   12              CONBAD
*
* EINGABE-/RUECKGABEPARAMETER
*
YDDXLOGL DS    H              LAENGE DER LOGON MESSAGE
YDDXLOGM DS    CL80           LOGON MESSAGE
*
* RUECKGABEPARAMETER
*
YDDXPW  DS    XL4           PASSWORT
YDDXRETC DS    F              RETURNCODE '..FDBKRC'
*
YDDXL   EQU   *-YDDEXPL     LAENGE DER DCAM EXIT PARAMETERLISTE
               *,YDDEXPL    060    950330   55647211
```

Description of the YDDEXPL fields

YDDXDPAR A(DARPPAR)
Address of an internal DCAM work area, via which further call-specific information is passed on to the exit routine.

YDDXMAIN Indicator for the main case

YDDXMAIN Indicator for the main case
YDDXAPPL main case APP (re: applications)
YDDXCONN main case CON (re: connections)

YDDXSUB Indicator for the subcase
YDDXAOPN exit case APPOPN
YDDXACLS exit case APPCLS
YDDXASHU exit case APPSHU
YDDXATRM exit case APPTRM
YDDXAFCL exit case APPFCL
YDDXCACQ exit case CONACQ
YDDXCACC exit case CONACC
YDDXCCPL exit case CONCPL
YDDXCREQ exit case CONREQ
YDDXCCLS exit case CONCLS
YDDXCFCL exit case CONFCL
YDDXCBAD exit case CONBAD

YDDXLOGL Length of the connection message (LOGON message) in field YDDXLOGLM. It must not exceed 80 bytes. This field can be modified by the exit routine.
YDDXLOGL = 0 means that there is no connection message.

YDDXLOGM Contains the connection message (LOGON message) which can be modified by the exit routine.

YDDXPW Contains a password, which the exit routine transfers to DCAM on returning. This password replaces the RDF password specified by the user for YOPEN or the LOGON password for YOPNCON ACQUIRE.

YDDXRETC Return code of the exit routine.

Notes on programming DCAM exit routines

The following restrictions apply to DCAM exit routines in addition to the rules specified in section “Exit routines” on page 10:

– All DCAM data structure fields (DARPPAR, control blocks, see below) are read-only. Write access is permitted only for return fields in parameter area YDDEXPL.
– No BCAM calls are permitted.
– The exit routine must not set any bourse locks.
– The exit routine must not change your task (PCB) level.

Excerpt from the DCAM data structure

Those DCAM data structure elements which are of greatest significance with regard to programming exit routines are described below.

Since compatibility for the layout of the DCAM control blocks is not guaranteed, the DCAM data structure can only be accessed via symbolic field names. Recompilation and source changes may prove necessary in the event of a DCAM version change.

Internal DCAM work area

DARPPAR contains task-specific data, e.g. the current addresses of the DCAM control blocks (cf. the “DCAM Macros” manual [1]).

A DSECT for DARPPAR is generated by the macro:

```
[name]  YDDDVEC  D [,prefix]
```

Default values:		
	name	YDDDVEC
	prefix	YDDV

DCAM application control block

DACB contains application-specific data, e.g. application name, attributes,... . The DACB is created on YOPEN and remains in existence until the application is closed.

A DSECT is generated by means of the macro:

```
[name] YDDDACB D [,prefix]
```

Default values:		
	name	YDDDACB
	prefix	YDDH

DCAM connection control block

DCCB contains connection-specific data, e.g. partner name, processor name. One DCCB is set up for each connection.

A DSECT is generated by means of the macro:
`[name] YDDDCCB D [,prefix]`

Default values: name YDDDCCB
prefix YDDL

DCAM request parameter block

DRPB contains the call-specific information from the RPB in the case of asynchronous calls.

A DSECT is generated by means of the macro:
`[name] YDDDRPB D [,prefix]`

Default values: name YDDDRPB
prefix YDDD

Field contents when calling exit routines

The contents of the fields of the exit parameter area and of the internal work area shown in the following table are dependent on the exit case. Fields and addresses which are not set have the default value X'00'.

YDDEXPL - exit parameter area

Field name	Field contents	Exit case											
		APP					CON						
		OPN	CLS	SHU	TRM	FCL	ACQ	ACC	CPL	REQ	CLS	FCL	BAD
YDDXMAIN	Exit main case	x	x	x	x	x	x	x	x	x	x	x	x
YDDXSUB	Exit subcase	x	x	x	x	x	x	x	x	x	x	x	x
YDDXDPAR	A(DARPPAR)	x	x	x	x	x	x	x	x	x	x	x	x
YDDXLOGL	L of LOGON message						x	x	x	x			
YDDXLOGM	A (LOGON message)						x	x	x	x			

YDDDVEC - internal DCAM work area

Field name	Field contents	Exit case											
		APP					CON						
		OPN	CLS	SHU	TRM	FCL	ACQ	ACC	CPL	REQ	CLS	FCL	BAD
YDDVXTID	A(TID)	x	x	x	x	x	x	x	x	x	x	x	x
YDDVADAC	A(DACB)	x	x	x	x	x	x	x	x	x	x	x	x
YDDVADCC	A(DCCB)						x	x	x			x	x
YDDVADRP	A(DRPB)						x	x	x				
YDDVARPB	A(RPB						x	x	x		x		
YDDVAPCN	A(PCN) *)							x		x			
YDDVAPTN	A(PTN) *)							x		x			

*) If the exit is initiated by the CONREQ exit case, there is no connection control bloc (DCCB). The user can obtain information about the partner address by means of YDDVAPTN, but cannot make any modifications.

PCN	CL8	PROCESSOR NAME
PTN	CL8	PARTNER NAME

Return code for the exit routine

For the exit cases APPCLS, APPSHU, APPTRM, APPFCL, CONCPL, CONCLS, CONFCL and CONBAD, the exit routine must set the rightmost byte in the YDDXRETC field to zero (RC = X'00'). These exit cases only pass information to the exit routine; DCAM processing cannot be affected by the exit routine.

In the case of exit cases APPOPN, CONACQ, CONACC and CONREQ, the exit routine can influence DCAM processing via the return code in the YDDXRETC field. The table below shows which return codes can be set.

Return codes (1)

		Condition	Exit case			
			APPOPN	CONACQ	CONACC	CONREQ
YDDRETC	RC = 00		x	x	x	x
	RC = 04		x	x	x	x
	RC = 08		x	x	x	-
	FDBK	RC = 04	+	+	+	+
YDDEXPL	YDDXPW	RC = 08	*	*	-	-
	YDDXLOGL	RC = 08	-	*	*	-
	YDDXLOGM	YDDXLOGL > 0	-	*	*	-

x the value/field can be specified

- the value/field cannot be specified

\+ the field must be specified if the condition applies.

* if RC=08 is set, the field can be modified and returned to DCAM.

Cases marked with an asterisk (*) in the table above are explained in greater detail in the following table:

Return codes (2)

Exit case	Field in YDDEXPL	Modification
APPOPN	YDDXPW	In the event of a BCAM call the RDF password (USEPW parameter in ACB) is replaced by the password transferred. If YDDXPW = 0, the password remains unchanged.
CONACC CONACQ	YDDXLOGL	YDDXLOGL = 0 means the connection message remains unchanged. YDDXLOGL = X' 7FFF' means the connection message is deleted.
CONACC CONACQ	YDDXLOGM	The connection message (LOGON message) of the user (contents of AREA) is replaced by the transferred data prior to the BCAM call.
CONACQ	YDDXPW	The LOGON password (LOGPW parameter in CCB) is replaced prior to the BCAM call by the password transferred. If YDDXPW = X' 00' , the password is not replaced.

SYSCMD exit (080)

The program branches to this exit whenever a system command is input, provided the task has been activated.

Three cases of command input can be distinguished:

– the command is input interactively from a data display terminal

– the command is input from a cataloged file (SPOOLIN, ENTER-JOB, CALL-PROCEDURE) or in remote batch mode

– the command is issued via the MCLP interface from a TU program (see the “Executive Macros” manual [10]).

The SYSCMD exit routine enables the command flow of all active tasks during the OPEN session to be monitored.

The exit routine can:
– accept the input command without modifications (return code EX080CAC)
– reject the command (return code EX080CRJ)
– modify or replace the command (return code EX080CRO)
– insert one or more new commands (return codes EX080CRS/EX080CLR).

The return code of the exit routine must be stored in the EX080RC field of parameter area EX080. The address of the parameter area is passed to the exit routine in R1.

The following information is passed to the exit routine:

```
R1    = A(EX080 parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

Notes

– The contents of registers 12, 13 and 14 must not be destroyed by the exit routine.

– The SYSCMD exit routine should process commands in the same way, regardless of how they were entered.

– If the exit routine is deactivated while a command is replaced by several commands, the following dummy command record is sent to the user program: `/REMARK SYSCMD EXIT NO MORE ACTIVE DURING PROCESSING`

– If commands are skipped in a procedure by means of /SKIP-COMMANDS, the exit routine is called once for each command skipped. The same applies in the case of an error when the spin-off mechanism (search for STEP, LOGOFF, SET-JOB-STEP) causes commands to be skipped.

– Variable substitution has already been effected in CALL-PROC procedures when the exit is called.

– In the case of branch flags, the branch destination is stored twice.

– The information for the calling system module must be returned in the EX080RC field of the parameter area.

DSECT

```
         EX080 D
EX080PL  DSECT
*-
*-       INPUT  INFORMATION
*-
EX080IND DC    AL1(0)                  SOURCE OF COMMANDS
EX080TER EQU   1                       -CMD READ FROM A TERMINAL
EX080FIL EQU   2                       -CMD READ FROM A JCL FILE
EX080MCL EQU   4                       -CMD READ VIA MCLP
EX080TRN EQU   8                       -CMD READ VIA TRCMD          999
EX080BUF EQU   16                      -CMD FOUND IN BS2000 BUFFER  999
*-
EX080BIT DC    X'00'                   BIT_INFO                     900
EX080SDF EQU   X'80'                   -INPUT IN SDF STRUCTURED FORM002
EX080SD2 EQU   X'40'                   -INPUT IN NEW SDF STRUCT FORM120
EX080STR EQU   X'00'                   -INPUT IN STRING MODE        900
*-
EX080UNU DC    2X'00'                                               900
*-
EX080IR  DC    A(0)                    A (INPUT RECORD)
*-
*-       OUTPUT INFORMATION
*-
EX080RC  DC    AL1(0)                  SYSCMD EXIT'S RETURN CODE
EX80CAC  EQU   0                       -CMD REC ACCEPTED
EX80CRJ  EQU   4                       -CMD REC REJECTED
EX80CRO  EQU   8                       -CMD REC REPLACED BY ONE
EX80CRS  EQU   12                      -CMD REC REPLACED BY SEVERAL
EX80CLR  EQU   16                      -LAST CMD REC REPLACED
         DC    3X'00'                  UNUSED IN VER=710
*-
EX080RR  DC    A(0)                    A (RETURNED RECORD)
*-
*-       ADDITIONAL INPUT  INFORMATION
*-
EX080ILR DC    A(0)                    A (LOWER CASE INPUT RECORD)
EX080STD DC    A(0)                    A (STD FORM)
EX080PLL EQU   *-EX080PL               LENGTH OF SYSCMD EXIT  P/L
```

Meanings of the fields:

EX080IND Type of command input
- EX080TER command input from terminal
- EX080FIL command input from JCL file
- EX080MCL command input from user program via MCLP

EX080BIT Indicator for SDF
- EX080SDF input in SDF structure
- EX080STR input not in SDF structure

EX080IR Address of input command record

EX080RC Return code
- EX080CAC command accepted
- EX080CRJ command rejected
- EX080CRO command modified or replaced
- EX080CRS command replaced by several commands
- EX080CLR last command record replaced

EX080RR Address of command record returned to calling system component when command is modified or replaced

Meanings of the return codes in the EX080RC field

Return code X'04': command rejection (EX080RC=EX080CRJ)

In interactive mode, no system message is output to the terminal. The exit routine can, however, send a message to the user.

In the case of command input from a cataloged file (PROC A or PROC C or corresponding SDF commands), the following message is output:

```
/REMARK COMMAND REJECTED BY THE SYSTEM ADMINISTRATOR
```

In batch mode the same message is entered in the logging file.

Return code X'08': command replacement or modification (EX080RC=EX080CRO)

The exit routine itself must construct the new or modified command and pass the address of the command record to the calling system component via the EX080RR field of the parameter area.

Format of the command record (if not standardized SDF transfer area):

0	2	4 RL-1
RL		command with operands

where RL = record length

ISAM keys, continuation characters and symbolic parameters (&...) are not permitted in the command record. The record length may be modified.

– In interactive mode, no system message is output.

– In the case of command input from a cataloged file, the new or modified command is output or entered in the logging file.

– When the command is input via MCLP, the replacing command must also be legal for MCLP, otherwise return code X'14' (invalid command) is returned to the user program. No DO, CALL, LOAD, LOGOFF, ABEND, EXECUTE or corresponding SDF commands may be returned and no other command may be called via MCLP, since that would cause the user program to be unloaded immediately.
If an error occurs during processing of the new command, X'10' (illegal command) is reported to the user program.

Return codes X'0C' and X'10': replacement of one command by several (EX080RC=EX080CRS/EX080CLR)

If the exit routine returns a command record to the system (see RC = 08) and simultaneously sets the return code in the EX080RC field of the parameter list to X'0C' (= EX080CRS), it is given control again immediately and can pass the next command record. As of the second call after return code X'0C', the address of the input command record (field EX080IR in the parameter area) is set to zero. When the exit routine returns the last command, it must set the return code in the parameter area to X'10' (=EX080CLR).

– In interactive mode, no system message is output for the replacement of one command by several commands.

– In batch mode, the replacing commands are written to the logging file.

– When a command is entered via MCLP, the notes on return code X'08' must be given due consideration. The commands mentioned above must not be returned via a command sequence, since this would cause the user program to be unloaded immediately. If a buffer has been provided in the user program for storing the results of MCLP command processing, this buffer must be able to accommodate the results of the entire command sequence. It is to be expected that many user programs working with MCLP calls will not run correctly when the exit routine replaces one command by several commands.

SDF exits (080/081/082/083)

In systems with SDF (System Dialog Facility), three further exits are provided in addition to exit 080 (SYSCMD exit, see above):

Exit number	Entered via	Type of exit	When called
080	SYSCMD	Request	Prior to command processing (like exit 080 without SDF)
081	SYSCMD	Return	After processing of the command by the decoder, provided no syntax error has been detected.
082	SYSSTMT	Request	Prior to statement processing by the decoder.
083	SYSSTMT	Return	After processing of the statement by the decoder, provided no syntax error has been detected.

All four exits use parameter area EX080. The address of the parameter area is passed to the exit routine in register 1.

By means of the exits, the command can be accepted, rejected, modified or replaced.

The return code of the exit routine must be stored in field EX080RC of the parameter area.

Note

In order to be able to make effective use of the exits in an SDF environment, the user will require an in-depth knowledge of the interface. Lack of space precludes the inclusion of such detailed information in this manual. The required information can, however be found in the following manuals: “SDF-A” [11], “Introductory Guide to the SDF Dialog Interface” [12], “Commands” [8], “Introductory Guide to Systems Support” [7].

SYSCMD exit 080

The description given above in exit 080 applies, supplemented by the following:

– The exit routine is not activated in SDF guided dialog.

– Activation of the exit routine takes place regardless of whether the command was read by the decoder itself (EX080BUF) or transferred by TRCMD (EX080TRN; see extension in parameter area EX080).

– If an input is detected which is not intended for the current system syntax file, the exit routine will be activated even if the input belongs to a lower-ranking “command stream” and no continuation line was expected.

– If data records are read when a command was expected, the exit routine is not called, and the error handling facility is initiated.

– If end-of-file is detected during an attempt to read, the exit routine is not called, and the EOF-handling measures are performed.

SYSCMD exit 081

This exit can be used only in systems with SDF. When it is called, the command to be processed has been recognized as syntactically correct by the decoder. Parameter substitution and continuation handling are complete.

Under normal circumstances, the exit routine receives the command in the form of an input string. The address is in the EX080IR field in the parameter area.

If the command has been entered in the syntax file using the SDF-A statement //ADD-COMMAND with the operand specification IMPLEMENTOR=P2 (...,CALL=NEW,...), the parameter area will be used to pass on to the exit routine the address of the standardized transfer area. The exit routine must be properly prepared for both forms of transfer. A description of the standardized transfer area and of the //ADD-COMMAND statement, as well as other useful information, may be found by consulting the “SDF-A” manual [11].

The following items of information are passed on to the exit routine:

```
R1    = A(EX080 parameter area)
R12   = A(TPR program manager)
R13   = A(save area of calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

Notes

– If commands are skipped using /SKIP-COMMANDS, the exit routine is not given control until the branch address label is reached.

– In the case of branch flags, the branch destination is stored twice.

– In the event of an error, the exit routine is not called again until a command initiating restart (/SET-JOB-STEP) is issued.

SYSSTMT exit 082

This exit has the same function as regards inputs from SYSSTMT as exit 080 has for inputs from SYSCMD. The technical details and notes in the section on SYSCMD exit 080 also apply here.

SYSSTMT exit 083

This exit corresponds to exit number 081. All information applies equally to both exits.

Notes

– The contents of registers 12, 13 and 14 must not be destroyed by the exit routine.

– The SYSTAT exit routine should process commands in the same way, regardless of how they were entered.

– If the exit routine is deactivated while a statement is replaced by several commands, the following dummy command record is sent to the user program:
/REMARK SYSCMD EXIT NO MORE ACTIVE DURING PROCESSING

– A command or statement can only be modified by commands or statements defined in the system syntax file.

– Variable substitution has already been effected in procedures when the exit is called.

DSECT

```
         EX080 D
EX080PL  DSECT
*-
*-       INPUT  INFORMATION
*-
EX080IND DC    AL1(0)                  SOURCE OF COMMANDS
EX080TER EQU   1                       -CMD READ FROM A TERMINAL
EX080FIL EQU   2                       -CMD READ FROM A JCL FILE
EX080MCL EQU   4                       -CMD READ VIA MCLP
EX080TRN EQU   8                       -CMD READ VIA TRCMD          999
EX080BUF EQU   16                      -CMD FOUND IN BS2000 BUFFER  999
*-
EX080BIT DC    X'00'                   BIT_INFO                     900
EX080SDF EQU   X'80'                   -INPUT IN SDF STRUCTURED FORM002
EX080SD2 EQU   X'40'                   -INPUT IN NEW SDF STRUCT FORM120
EX080STR EQU   X'00'                   -INPUT IN STRING MODE        900
*-
EX080UNU DC    2X'00'                                               900
*-
EX080IR  DC    A(0)                    A (INPUT RECORD)
*-
*-       OUTPUT INFORMATION
*-
EX080RC  DC    AL1(0)                  SYSCMD EXIT'S RETURN CODE
EX80CAC  EQU   0                       -CMD REC ACCEPTED
EX80CRJ  EQU   4                       -CMD REC REJECTED
EX80CRO  EQU   8                       -CMD REC REPLACED BY ONE
EX80CRS  EQU   12                      -CMD REC REPLACED BY SEVERAL
EX80CLR  EQU   16                      -LAST CMD REC REPLACED
         DC    3X'00'                  UNUSED IN VER=710
*-
EX080RR  DC    A(0)                    A (RETURNED RECORD)
*-
*-       ADDITIONAL INPUT  INFORMATION
*-
EX080ILR DC    A(0)                    A (LOWER CASE INPUT RECORD)
EX080STD DC    A(0)                    A (STD FORM)
```

Meanings of the fields:

EX080IND Source of the command
- EX080TER Command input from the terminal
- EX080FIL Command input from a JCL file
- EX080MCL Command input via MCLP from a user program
- EX080TRN Command input by TRCMD
- EX080BUF Command read internally

EX080BIT Structure of the input
- EX080SDF Input in SDF format
- EX080STR Input not in SDF format

EX080IR Address of the input command record or of the standardized transfer area

EX080RC Return code of the system exit
- EX080CAC Command accepted
- EX080CRJ Command rejected
- EX080CRO Command modified or replaced
- EX080CRS Command replaced by several commands
- EX080CLR Last command record replaced

EX080RR Address of command record returned to calling system component upon modification or replacement.

EX080ILR Address of the input record

EX080STD Address of the standard input

Meanings of the return codes in the EX080RC field

Return code X'04': command rejected (EX080RC=EX080CRJ)

In interactive mode, no system message is output to the terminal. The exit routine can, however, send a message to the user.

In the case of command input from a cataloged file the following message is output:

```
/REMARK COMMAND REJECTED BY THE SYSTEM ADMINISTRATOR
```

In batch mode the same message is entered in the logging file.

Return code X'08': command replacement or modification (EX080RC=EX080CRO)

The exit routine must construct the new or modified command itself and supply values to the standardized transfer area (see the "SDF-A" manual [11]). The address of the command record (of the standardized transfer area) is passed to the calling system component via the EX080RR field of the parameter area.

Format of the command record (if not standardized SDF transfer area):

0	2	4 RL-1
RL		command with operands

where RL = record length

ISAM keys, continuation characters and symbolic parameters (&...) are not permitted in the command record. The record length may be modified.

In interactive mode, no system message is output.

In the case of command input from a cataloged file, the new or modified command is output or entered in the logging file.

When the command is input via MCLP, the replacing command must also be legal for MCLP, otherwise return code X'14' (invalid command) is returned to the user program.
No CALL-PROCEDURE, LOGOFF, LOAD-PROGRAM or START-PROGRAM command may be returned and no other command may be called via MCLP, since that would cause the user program to be unloaded immediately.
If an error occurs during processing of the new command, X'10' (illegal command) is reported to the user program.

Return codes X'0C' and X'10': replacement of one command by several (EX080RC=EX080CRS/EX080CLR)

If the exit routine returns a command record to the system (see RC=X'08') and simultaneously sets the return code in the EX080RC field of the parameter area to X'0C' (= EX080CRS), it is given control again immediately and can pass the next command record. As of the second call after return code X'0C', the address of the input command record (field EX080IR in the parameter area) is set to zero. When the exit routine returns the last command, it must set the return code in the parameter area to X'10'.

In interactive mode, no system message is output for the replacement of one command by several commands.

In batch mode, the replacing commands are written to the logging file.

When a command is entered via MCLP, the notes on return code X'08' must be taken into consideration. The commands mentioned above (CALL-PROGRAM etc.) must not be returned via a command sequence, since this would cause the user program to be unloaded immediately. If a buffer has been provided in the user program for storing the results of MCLP command processing, this buffer must be able to accommodate the results of the entire command sequence.

It is to be expected that many user programs working with MCLP calls will not run correctly when the exit routine replaces one command by several commands.

SPOOL exits (090/091/092/094/095/096/097)

General information on SPOOL exits

It is possible, for each individual output device, to define whether SPOOL outputs on this device are to be processed using exit routines (see the START-PRINTER-OUTPUT command).

Querying the current SPOOL version

The SPOOL (and RSO) exit routines must query the numbers of the current SPOOL version in the SCB.
Exit routines are loaded, activated and deactivated independently of the DSSM commands START-SUBSYSTEM, STOP-SUBSYSTEM, HOLD-SUBSYSTEM and RESUME-SUBSYSTEM. As a result, SPOOL can be unloaded and replaced by another version, while the exit routines remain activated and can be called later by the newly loaded SPOOL version.
The exit routines have to be able to recognize this change to the SPOOL version; to this end they check the version specification stored in the SCB.

The version number is contained in the SCBESCB field of the DSECT, which can be created using the SSVSCB macro.

If system exits 90, 91 and 92 are active at the same time, the calls occur in the following order:

Exit 90: First call

Exit 91: Header page routine

Exit 90: Processing routine

Exit 92: Trailer page routine

Exit 90: Final call.

The first call has already been processed when the file is opened. The next step is to format the header page.
Once the trailer page has been generated, the final call is issued and the file is closed.

Independently of this sequence, system exit 95 may have been activated in order to monitor channel programs sent to the printers.

In the case of “family” processing, a header can be generated either for the first file only or for every file in the family. By means of an SPSINF call (INFO=SPSGEN) in the exit routine you can read the GEN entry and thus obtain the information you require.
If GENFAM=YES is set in the GEN entry, a header page is printed for each file in the “family” printout; otherwise only a header page for the first file will be printed (i.e. if the field SCBESEQ#=0 is set in the SCB).

Systems support staff can use exit 094 to generate a user routine for assistance in controlling the conversion of resources within a client/server environment in conjunction with the product Distributed Print Services (Dprint).

Using exit 096 in conjunction with the product Distributed Print Services enables systems support staff to select particular servers by combining the appropriate selection criteria.

Exit 097 can be used to control the number of jobs of the product Dprint (Distributed Print Services) for file transfer.

Note

The descriptions on the following pages refer to the new PRINT-DOCUMENT command or PRNTDOC macro introduced with SPOOL V3.0. The exit functions can be used in the same way with the old PRINT-FILE command or the old PRNT macro. More detailed information on SPOOL can be found in the manual “SPOOL” [14]. For Dprint exits please use the description given in the manual “Distributed Print Services” [5].

SPOOL exit (090)

This exit routine is called after a record from the output file has been read, but before the record is subjected to further processing by SPOOL in accordance with the entries in the PRINT-DOCUMENT command.

This exit may be used for output via line printer or laser printer (page printer). It cannot be used for output to magnetic tape or floppy disk.

The exit routine can initiate the following system functions:

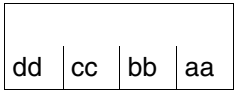
– accept the record that has been read in (updated or not updated)
– replace the record that has been read in
– skip the record that has been read in
– skip the record that has been read in and all subsequent records up to the end of the file.

The following information is passed to the exit routine:

```
R1    = A(EX090 parameter area)
R12   = A(TPR program manager)
R13   = A(save area)
R14   = A(indirect return)
R15   = A(exit routine)
```

The return codes are entered in the standard header.

Standard header:



A return code concerning execution of EX090 is passed in the standard header.
aa=maincode1; bb=maincode2; cc=subcode1; dd=subcode2

X'aa'	Meaning
00	Accept the record that has been read in
04	Replace the record that has been read in with the contents of the output block
08	Skip the record that has been read in
0C	Skip the record that has been read in and all following records up to the end of the file

Notes

– The exit routine should be neither activated nor deactivated while a spoolout is being processed.

– The PRNTDOC macro may be utilized in the exit routine, but not $PRNT and $SPRQ.

– Control is returned to the exit routine for every record read from the input file (including for the sample printout of the first page), until the operator initiates printing. The same applies to RESPOOL processing (HOLD-PRINT-JOB/RESUME-PRINT-JOB command), where the exit routine must also take account of the location of the start of RESPOOL.

– The information required for processing files with non-Siemens control characters can be found in the PRINT-DOCUMENT command and the SPOOL control block.

– If output blocks are provided, any memory class is permissible. However, it is advisable to choose class 5 memory. The memory is then released by the exit as soon as the exit has regained control.

DSECT

A DSECT/CSECT for the parameter area (the address is transferred in register 1) can be generated by means of the EX090 macro (“prefix” must not be longer than 3 characters):

```
name EX090 D/C ,[prefix]
```

After the return has taken place, the input fields are subject to no more checks on the part of the calling SPOOL component. The first and last exit routine calls take place without a record from the file being made available (ICLTYP = X’02’ or X’03’).
The byte IPRTYP specifies the output device.

Values for the output fields are supplied by the exit routine.

The byte IEXRC controls the function required by the exit routine. A more detailed analysis of the functions follows the DSECT.

```
          EX090 D
* SPOOL INPUT RECORDS ROUTINE EXIT PARAMETER LIST
*
IRC       MFCHK MF=D,DMACID=E90,ALIGN=F,SUPPORT=(C,D),ENTRY=NO,         C
                PREFIX=I,MACID=E90,DNAME=C
IRC       DSECT ,
                *,##### PREFIX=I, MACID=E90 #####
*
*   STANDARD HEADER RETURN INFORMATION
*
IBEG      DS    0H
00100510
          FHDR  MF=(C,I),EQUATES=NO
00100600
          DS    0A
IFHE      DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
IIFID     DS    0A              0   INTERFACE IDENTIFIER
IFCTU     DS    AL2             0   FUNCTION UNIT NUMBER
*                                   BIT 15    HEADER FLAG BIT,
*                                   MUST BE RESET UNTIL FURTHER NOTICE
*                                   BIT 14-12 UNUSED, MUST BE RESET
*                                   BIT 11-0  REAL FUNCTION UNIT NUMBER
IFCT      DS    AL1             2   FUNCTION NUMBER
IFCTV     DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
IRET      DS    0A              4   GENERAL RETURN CODE
ISRET     DS    0AL2            4   SUB RETURN CODE
ISR2      DS    AL1             4   SUB RETURN CODE 2
ISR1      DS    AL1             5   SUB RETURN CODE 1
IMRET     DS    0AL2            6   MAIN RETURN CODE
IMR2      DS    AL1             6   MAIN RETURN CODE 2
IMR1      DS    AL1             7   MAIN RETURN CODE 1
IFHL      EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*   EQUATES FOR EXIT RETURN INFORMATION
*
IACCPT    EQU   0                       ACCEPT RECORD AS IT IS
IADD      EQU   4                       ADD RECORD(S)
IDEL      EQU   8                       DELETE (SKIP) RECORD
IEOF      EQU   12                      TERMINATE PRINT (SIMULATE EOF)
```

```
*   INPUT FIELDS FOR THE INPUT EXIT ROUTINE
*
IREC@    DS    A                       INPUT RECORD ADDRESS
ISCB@    DS    A                       SCB ADDRESS
IRCLEN   DS    Y                       INPUT RECORD LENGTH
IPRTYP   DS    AL1                     TYPE OF PRINT
ILINES   EQU   32                     LINES PRINTER TYPE           751
IPAGES   EQU   33                     PAGE PRINTER TYPE            751
ILOADL   EQU   34                     LOADABLE LINE PRINTER TYPE   751
IRBP     EQU   35                     RBP PRINTER TYPE             751
IRSOPB   EQU   36                     PUBLIC RSO PRINTER           752
IRSOPR   EQU   37                     PRIVATE RSO PRINTER          752
IAPAPR   EQU   38                     APA PRINTERS
IPUNCH   EQU   48                     PUNCH TYPE                   751
ICLTYP   DS    AL1                     CALL TYPE
ICLREC   EQU   1                       RECORD CALL TYPE
ICLFST   EQU   2                       NO RECORD FIRST TIME CALL
ICLLST   EQU   3                       NO RECORD LAST TIME CALL
         DS    CL4                     RESERVED FOR EXTENSION
*
*   OUTPUT FIELDS FROM THE INPUT EXIT ROUTINE
*
IBLK@    DS    A                       DATA BLOCK ADDRESS
IBLKLE   DS    Y                       DATA BLOCK LENGTH
IEXRC    DS    AL1                     R-C FROM EXIT ROUTINE MAY BE
*                                      STORED HERE IN ADDITION TO
*                                      THE MAIN RETURN CODE
IBEXRC   DS    AL1                     BASE R-C FROM EXIT ROUTINE
*
IBRCOK   EQU   0                       BASE R-C O.K.
INEXS    EQU   4                       EXIT ROUTINE NOT EXISTENT
INACT    EQU   8                       EXIT ROUTINE NOT ACTIVE
IPARER   EQU   12                      PARAMETER ERROR
         DS    CL8                     RESERVED FOR EXTENSION
*
*   INPUT FIELDS FOR THE INPUT EXIT ROUTINE
*
IPIR     DS    A                 PRINTER INFORMATION RECORD ADDR.  755
IDVTYP   DS    AL1               DEVICE TYPE                       755
         DS    CL11              RESERVED FOR EXTENTION            755
ILEN     EQU   *-IRC                 ROUTINE EXIT P/L LENGTH
```

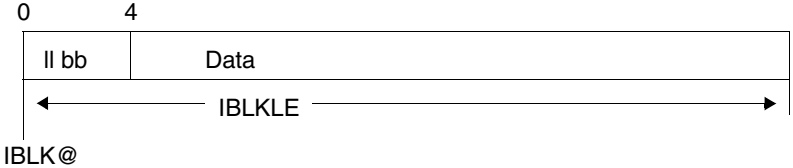
Exit routine functions

Return code X'00': accept the record that has been read

The address of the record and the record length are passed to the exit routine in the IREC@ or IRCLEN field. The record contains no record length field.

The exit routine can have SPOOL accept the record in unmodified form, or modify it itself. If its length is not increased at this stage, the record can be modified directly in the input buffer. In this case, the fields IBLK@ and IBLKLE must be supplied with the address of the record from IREC@ and the correct record length. If the modification increases the length of the record, then the exit routine must make the record available in a special buffer, addressed via IBLK@:

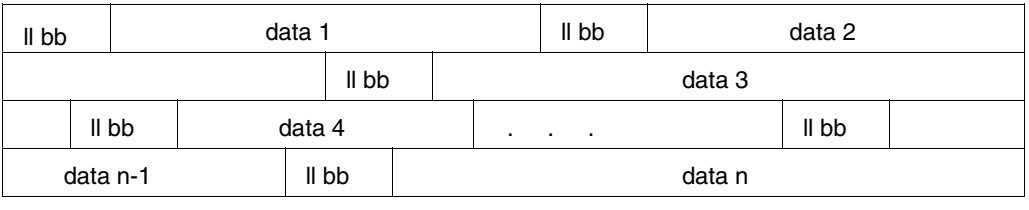
Buffer format:



ll data length + 4 (IBLKLE)
bb blanks

Return code X'04': replace the record that has been read

SPOOL replaces the record which has been read in by those provided by the exit routine in the output block (address in IBLK@):



The exit routine must supply the contents for fields IBLK@ and IBLKLE.

If the record read is to be a component of the output block, the exit routine must transfer it to this block.

Return code X'08': skip the record that has been read

The exit routine needs only to set the return code. No other action is necessary.

Return code X'0C': skip all records up to the end of the file

The exit routine needs only to set the return code. No further action is necessary.

Operands in the PRINT-DOCUMENT command and PRNTDOC macro

The exit routine must take into consideration all operands specified in PRINT-DOCUMENT commands. The appropriate values can be derived from the SPOOL control block.

*RECORD-PART=*PARAMETERS(FIRST-CHARACTER = n)*:
All records inserted by the exit routine must be longer than n bytes, otherwise they are not printed.

*OUTPUT-PART=*RANGE(FROM = +n)*:
Records 1 to n are transferred to the exit routine.

*OUTPUT-PART=*LAST(LAST = -n)*:
Initially, the exit routine has control for all records from 1 to m (end-of-file, no output). SPOOL then sets the FROM value to z=(m-n+1) and processes the file as for FROM = +z.

ADDITIONAL-COPIES = n:
In the case of multiple output of a file, the exit routine cannot distinguish between the individual copies, i.e. there is no difference in the parameters or data records transferred.

SPOOL exits for header and trailer pages (091/092)

These system exits are used to control the layout of header/trailer pages for printouts.

In particular, the exit routines can be used to suppress the standard output of the SPOOL system, replace it by one of several alternative outputs, or initiate an alternative output in addition to the standard output.

Alternative outputs for header or trailer pages are made available by systems support. They can be requested by the user by means of the HEADER-EXIT-NUMBER and TRAILER-EXIT-NUMBER operands in the PRINT-DOCUMENT command. The standard output can be modified by a user text (HEADER-PAGE-TEXT operand in the PRINT-DOCUMENT command) independently of an exit routine.

Example

A standard name must be defined, e.g.

```
HEADERPAGETEXT.n,
```

This name denotes a SAM or ISAM file or a job variable. It can be used to store texts, which in turn can be accessed using the command:

```
PRINT-DOCUMENT ...,LAYOUT-CONTROL=*PARAMETERS(COVER-PAGES= -
  *PARAMETERS(HEADER-EXIT-NUMBER=...))
```

The exit routine must access the file: `$userid.HEADERPAGETEXT.n`

and enter the text in the buffer for the header page. No selection is made until the exit routine has been called. These exit routines save and restore all registers.

The routines are called:

– if no header/trailer page was created

– to restore the management buffer in cases where a header/trailer page already exists

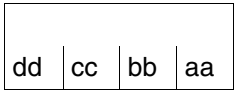
Header page exit (091)

The following information is passed to the exit routine:

```
R1    = A(EX091 parameter area)
R12   = A(TPR program manager)
R13   = A(save area)
R14   = A(indirect return)
R15   = A(exit routine)
```

The return codes are entered in the standard header.

Standard header:



A return code concerning execution of EX091 and EX092 is passed in the standard header.
aa=maincode1; bb=maincode2; cc=subcode1; dd=subcode2

X'aa'	**Meaning**
00	No header/trailer page
01	Print system header/trailer page without changes
02	Print specified alternative header/trailer page
03	Print the system header/trailer page without changes and the specified alternative header/trailer page
04	Only for exit 92. No trailer page. The last feed to vertical tab “channel 1” should be suppressed

DSECT

A DSECT or CSECT for the parameter area (address transferred in register 1) can be generated by means of the EX091 macro (“prefix” must not be longer than 3 characters):

```
         EX091 D
*
*   SPOOL HEADER  PAGE  ROUTINE EXIT PARAMETER LIST
*
IHD      MFCHK MF=D,DMACID=E91,ALIGN=F,SUPPORT=(C,D),ENTRY=NO,         C
               PREFIX=I,MACID=E91,DNAME=C
IHD      DSECT ,
               *,##### PREFIX=I, MACID=E91 #####
*
*   STANDARD HEADER RETURN INFORMATION
*
IBEG     DS    0H
         FHDR  MF=(C,I),EQUATES=NO
         DS    0A
IFHE     DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
IIFID    DS    0A              0   INTERFACE IDENTIFIER
IFCTU    DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
IFCT     DS    AL1             2   FUNCTION NUMBER
IFCTV    DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
IRET     DS    0A              4   GENERAL RETURN CODE
ISRET    DS    0AL2            4   SUB RETURN CODE
ISR2     DS    AL1             4   SUB RETURN CODE 2
ISR1     DS    AL1             5   SUB RETURN CODE 1
IMRET    DS    0AL2            6   MAIN RETURN CODE
IMR2     DS    AL1             6   MAIN RETURN CODE 2
IMR1     DS    AL1             7   MAIN RETURN CODE 1
IFHL     EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*
*   EQUATES FOR EXIT RETURN INFORMATION IN &P.MR1
*
IEXNH    EQU   0                       NO HEADER PAGE OUTPUT
IEXSY    EQU   1                       SYSTEM HEADER UNALTERED
IEXAL    EQU   2                       ALTERNATIVE HEADER
IEXBO    EQU   3                       SYSTEM AND ALTERNATIVE HEADER
*
```

```
*   INPUT FIELDS FOR THE INPUT EXIT ROUTINE
*
IPRTYP   DS    C                    PRINTER TYPE
ILINES   EQU   X'20'                LINES PRINTER TYPE
IPAGES   EQU   X'21'                PAGE PRINTER TYPE
ILOADL   EQU   X'22'                LOADABLE LINE PRINTER TYPE    753
IRBP     EQU   X'23'                RBP PRINTER TYPE              753
IRSOPB   EQU   X'24'                PUBLIC RSO PRINTER            753
IRSOPR   EQU   X'25'                PRIVATE RSO PRINTER           753
IAPAPR   EQU   X'26'                APA PRINTER
IPUNCH   EQU   X'30'                PUNCH TYPE                    753
ICLTYP   DS    C                    CALL TYPE
INORM    EQU   X'01'                NORMAL CALL                   754
IRETRY   EQU   X'02'                RETRY                         754
INOBUF   EQU   X'04'                NO BUFFER                     754
ISMLHD   EQU   X'08'                SMALL HEADER                  802
         DS    CL2                  RESERVED
ISCB@    DS    A                    SCB ADDRESS
ISYIB@   DS    A                    SYSTEM BUFFER
ILNPG    DS    F                    # OF LINES PER PAGE
ICHLN    DS    F                    # OF CHAR. PER LINE
*
*   OUTPUT FIELDS FROM THE INPUT EXIT ROUTINE
*
IUSOB@   DS    A                    USER BUFFER ADDRESS
IUSOBL   DS    F                    USER BUFFER LENGTH
ISKIP#   DS    F                    # SKIP RECORDS
ISYDF    DS    F                    SYSTEM DUPL FACT
IUSDF    DS    F                    USER DUPL FACT
         DS    CL12                 RESERVED FOR EXTENSION
*
*   INPUT FIELDS FOR THE INPUT EXIT ROUTINE
*
IPIR     DS    A                 PRINTER INFORMATION RECORD ADDR. 756
IDVTYP   DS    AL1               DEVICE TYPE                      756
         DS    XL1               HW ALIGNMENT
IFREQ#   DS    H                 SEPARATOR PAGES FREQUENCE #      865
*                                FREQ# IS INPUT / OUTPUT          865
IFWFR@   DS    A                 ADDR OF 10 FREE FW FOR EXIT 091  865
         DS    2F                RESERVED - FREE FOR USE          865
ILEN     EQU   *-IHD                   ROUTINE EXIT P/L LENGTH
```

Field contents when calling the exit routine:

IPRTYP	Output device	
	ILINES	line printer
	IPAGES	laser printer / page printer
	ILOADL	printer with loadable VFB
	IRSOPB	RSO public printer
	IRSOPR	RSO private printer
	IAPAPR	APA printer
ICLTYP	Type of call	
	INORM	normal call
	IRETRY	retry: required when the memory area for the alternative header page has been destroyed or not set up correctly.
	INOBUF	no system standard header page: no buffer address ISYIB@ is transferred. The exit routine can initiate an alternative output (ILNPG and ICHLN, see below).
	FSMLHD	short header page (after a job is aborted)
ISCB@	Address of SPOOL control block (SCB).	
ISYIB@	Address of system buffer.	
ILNPG	Number of lines per header page.	
ICHLN	Number of characters per line on header page.	

The exit routine can supply values to the following fields:

IUSOB@ Class 3 memory address at which the alternative header page is available.

IUSOBL Buffer length used for the alternative header page corresponding to the buffer area requested by the exit routine with $GETMEM.

ISKIP# Number of records in the alternative header page which are to be skipped, i.e. not printed, in the event of a retry.

ISYDF Iteration factor for the output of the standard header page. Default value: 1.

IUSDF Iteration factor for the output of the alternative header page.
Default value: 1.

IPIR Address of the special printer information record.
For more details see the relevant hardware description. (Only LP65 printers are currently supported.)

IDVTYP Device type
For more details see the description of the SPOOL macro SPSINF.

Trailer page exit (092)

The exit routine is provided with the following information:

```
R1    = A(EX092 parameter area)
R12   = A(TPR program manager)
R13   = A(save area)
R14   = A(indirect return)
R15   = A(exit routine)
```

DSECT

A DSECT or CSECT for the parameter area (address in R1) can be generated by means of the EX092 macro (“prefix” must not be longer than 3 characters):

```
          EX092 D
*
*   SPOOL TRAILER PAGE  ROUTINE EXIT PARAMETER LIST
*
ITR      MFCHK MF=D,DMACID=E92,ALIGN=F,SUPPORT=(C,D),ENTRY=NO,         C
               PREFIX=I,MACID=E92,DNAME=C
ITR      DSECT ,
               *,##### PREFIX=I, MACID=E92 #####
*
*   STANDARD HEADER RETURN INFORMATION
*
IBEG     DS    0H
         FHDR  MF=(C,I),EQUATES=NO
         DS    0A
IFHE     DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
IIFID    DS    0A              0   INTERFACE IDENTIFIER
IFCTU    DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
IFCT     DS    AL1             2   FUNCTION NUMBER
IFCTV    DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
IRET     DS    0A              4   GENERAL RETURN CODE
ISRET    DS    0AL2            4   SUB RETURN CODE
ISR2     DS    AL1             4   SUB RETURN CODE 2
ISR1     DS    AL1             5   SUB RETURN CODE 1
IMRET    DS    0AL2            6   MAIN RETURN CODE
IMR2     DS    AL1             6   MAIN RETURN CODE 2
IMR1     DS    AL1             7   MAIN RETURN CODE 1
IFHL     EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
```

```
*
*
*   EQUATES FOR EXIT RETURN INFORMATION IN &P.MR1
*
IEXNT    EQU   0                        NO TRAIL. PAGE OUTPUT
IEXSY    EQU   1                        SYSTEM TRAIL. UNALTERED
IEXAL    EQU   2                        ALTERNATIVE TRAILER PAGE
IEXBO    EQU   3                        SYSTEM AND ALTERNATIVE TRAILER
*   INPUT FIELDS FOR THE INPUT EXIT ROUTINE
*
IPRTYP   DS    C                        PRINTER TYPE
ILINES   EQU   X'20'                  LINES PRINTER TYPE
IPAGES   EQU   X'21'                  PAGE PRINTER TYPE              753
ILOADL   EQU   X'22'                  LOADABLE LINE PRINTER TYPE     753
IRBP     EQU   X'23'                  RBP PRINTER TYPE               753
IRSOPB   EQU   X'24'                  PUBLIC RSO PRINTER             753
IRSOPR   EQU   X'25'                  PRIVATE RSO PRINTER            753
IAPAPR   EQU   X'26'                  APA PRINTERS
IPUNCH   EQU   X'30'                  PUNCH TYPE                     753
ICLTYP   DS    C                        CALL TYPE
INORM    EQU   X'01'                  NORMAL CALL                    754
IRETRY   EQU   X'02'                  RETRY                          754
INOBUF   EQU   X'04'                  NO BUFFER                      754
         DS    CL2                RESERVED
ISCB@    DS    A                        SCB ADDRESS
ISYIB@   DS    A                        SYSTEM BUFFER
ILNPG    DS    F                  # OF LINES PER PAGE
ICHLN    DS    F                        # OF CHAR. PER LINE
*
*   OUTPUT FIELDS FROM THE INPUT EXIT ROUTINE
*
IUSOB@   DS    A                        USER BUFFER ADDRESS
IUSOBL   DS    F                        USER BUFFER LENGTH
ISKIP#   DS    F                  # SKIP RECORDS
ISYDF    DS    F                  SYSTEM DUPL FACT
IUSDF    DS    F                  USER DUPL FACT
         DS    CL12                     RESERVED FOR EXTENSION
*
*   INPUT FIELDS FOR THE INPUT EXIT ROUTINE
*
IPIR     DS    A                  PRINTER INFORMATION RECORD ADDR. 756
IDVTYP   DS    AL1                DEVICE TYPE                      756
         DS    CL11               RESERVED FOR EXTENTION           756
ILEN     EQU   *-ITR                   ROUTINE EXIT P/L LENGTH
```

Field contents when calling the exit routine:

YDDXMAIN Indicator for the main case
YDDXAPPL main case 'APP' (re: applications))

IPRTYP Output device
ILINES line printer
IPAGES laser printer / page printer
ILOADL printer with loadable VFB
IRBP RBP printer
IRSOPB RSO public printer
IRSOPR RSO private printer
IAPAPR APA printer

ICLTYP Type of call
INORM normal call
IRETRY retry: required when the memory area for the alternative trailer page has been destroyed or not set up correctly.
INOBUF no system standard trailer page: no buffer address ISYIB@ is transferred. The exit routine can initiate an alternative output (ILNPG and ICHLN, see below).
FSMLHD short trailer page (after a job is aborted)

ISCB@ Address of SPOOL control block (SCB).
ISYIB@ Address of system buffer.
ILNPG Number of lines per trailer page.
ICHLN Number of characters per line on trailer page.

The exit routine can supply values to the following fields:

IUSOB@ Class 3 memory address at which the alternative trailer page is available.

IUSOBL Buffer length used for the alternative trailer page corresponding to the buffer area requested by the exit routine with $GETMEM.

ISKIP# Number of records in the alternative trailer page which are to be skipped, i.e. not printed, in the event of a retry.

ISYDF Iteration factor for the output of the standard trailer page.
Default value: 1.

IUSDF Iteration factor for the output of the alternative trailer page.
Default value: 1.

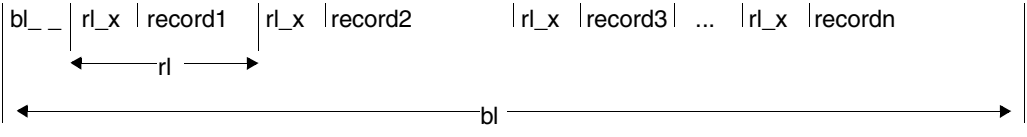
IPIR Address of the special printer information record.
For more details see the relevant hardware description. (Only LP65 printers are currently supported.)

IDVTYP Device type
For more details see the description of the SPOOL macro SPSINF.

Structure of the buffer for header/trailer pages

Alternative output

In order to set up an alternative output, the exit routine must reserve a buffer in class 3 memory. The header/trailer pages are made up of records of variable length, i.e. prefixed by a 4-byte record length field. The first 4 bytes of the buffer contain, left-justified, the buffer length used (maximum length: 32767 bytes = X'7FFF').



where:

bl buffer length used
rl record length/line length + 4
_ blank (X' 40')
x code character for type of output (only evaluated internally by SPOOL)
X'0A' printer

record for PRINT output: 1 byte print control character (see table) + text

Control char.	Meaning
X' 40'	No feed before printing
X' 41'	1 line feed before printing
X' 42'	2 line feeds before printing
X' 43'	3 line feeds before printing

Other control characters may lead to undesired results, as they are not interpreted by every device type.

System header page

The header page format depends on the size of the paper and the character set used. The number of lines which can be output on one printed page is determined as follows:

```
Maximum number of lines = Paper format in inches * lines / inches
```

Exception: in the case of laser printers, if the header page is output under LOOP control, the number of lines is preset by the VFB.

The lines of the header page are printed according to the order of priority described below (depending on the maximum number of lines).

Order of priority:

```
Address and identification field (24 lines)
User text (10 lines) + 2 blank lines          )
Job name (10 lines) + 2 blank lines           )in "oversize"
User ID (10 lines) + 2 blank lines            ) letters
Account number (10 lines) + 2 blank lines     )
```

If fewer than 24 lines are available, no standard header page is printed. The exit routine can, however, take appropriate action.

Each print line is stored as a variable-length record, i.e. with a leading record length field (4 bytes). Blank lines contain a blank (X'40') as the record contents.

Example

Format of a standard header page, contingent upon character and line density

Format (page size): 9 x 11 inches

Character set A: 15 characters/inch; 8 lines/inch (= 15 cpi / 8 lpi)

Character set B: 12 characters/inch; 6 lines/inch (= 12 cpi / 6 lpi)

	Character set A (15 cpi / 8 lpi)	**Character set B (12 cpi / 6 lpi)**
Number of characters per line	11 * 15 = 165	11 * 12 = 132
Number of lines per page	9 * 8 = 72	9 * 6 = 54
Header page format:		
Address field (24 lines) Job name (10 + 2 lines) User ID (10 + 2 lines) Account number (10 + 2 lines)	blank lines: 48 36 24 12	blank lines: 30 18 6 -
Comments	The entire header page is printed out	Account number missing
Length of asterisk lines	165 '*'	132 '*'

Paper feed

After the system trailer page has been printed, the next page is positioned to the first line. This also applies if the header/trailer page is not printed.

Notes on application

The optimum exploitation of the exits described above assumes appropriate cooperation between systems support and the customers, e.g. a standard name for a SAM or ISAM file (or even a job variable). Both privileged and nonprivileged users can use this file to set up an alternative header page.

Example of the format of a standard name:

```
HEADERPAGETEXT.n, 0 ≤ n ≤ 2147.483.639
```

The alternative header page can be printed using the PRINT-DOCUMENT command (operand HEADER-EXIT-NUMBER=n) or PRNTDOC macro (operand COVPAGE).

The exit routine must then access the file $userid.HEADERPAGETEXT.n and provide the header page buffer with the alternative text.

Whether a header/trailer page is printed depends on the number of characters that can be printed per line.

If separate output blocks are provided by exits 091/092, class 3 memory is required. The memory is released by SPOOL, not by the exits.

SPOOL exit 094 (resources routine)

This SPOOL exit enables systems support to create a user routine designed to carry out two functions within a client/server environment in conjunction with the product Distributed Print Services.

Conversion of client resources to server resources (function F0)

This function modifies the values of certain fields in the SCB (e.g. FORM, DIA).
Input consists of the standard header, the address of the SCB and function code X'F0' or C'0' in the E94INFOR field.
Output consists of the return code in the standard header and the modified SCB.

Conversion of server resources to client resources (function F1)

This function restores the original values of the SCB fields (e.g. FORM, DIA). Input consists of the standard header, the address of the SCB and function code X'F1' or C'1' in the E94INFOR field.
Output consists of the return code in the standard header and the modified SCB.

Exit 094 must be started on both the client and the server. If this is not done, the spoolout request is rejected with the following message:

```
%  SCP1089 EXIT#094 MANDATORY FOR TSN '(&00)'. COMMAND REJECTED
```

Use of the exit routine with function code F1 in order to restore the original name in the SCB is optional. Only the return code has to be set. However, even in this case the conversion performed by the server is transparent for the user on the client. Output has been modified accordingly for all SHOW-PRINT-JOB commands. The cluster administrator is always shown the converted values.

By default, the trailer page is printed with the values of the server. Users wishing to print a trailer page with the values of the client must create an appropriate trailer exit by calling exit 094.

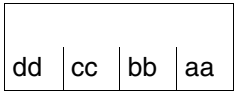
If an error occurs while processing function call F0 (return code not equal to zero in the standard header), the spoolout request is rejected with message SCP1090.

If an error occurs while processing function call F1 (return code not equal to zero in the standard header), an error message is displayed at the console, but the spoolout job is continued.

The following information is passed to the exit routine:
`R1 = A (EX094 parameter area)`

The return codes are entered in the standard header.

Standard header:



A return code concerning execution of EX094 is passed in the standard header.
aa=maincode1; bb=maincode2; cc=subcode1; dd=subcode2

X'aa'	**Meaning**
00	Exit processed without errors
01	Exit not processed without errors

DSECT

A DSECT/CSECT for the parameter area can be generated using the EX094 macro. “prefix” must not be longer than 3 characters.

```
         EX094 D
CLREXPL  DSECT ,              RENAMING OF CLIENT RESOURCE EXIT P/L
E94EXBEG DS    0H
         FHDR  MF=(C,E94),EQUATES=NO
         DS    0A
E94FHE   DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
E94IFID  DS    0A              0   INTERFACE IDENTIFIER
E94FCTU  DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
E94FCT   DS    AL1             2   FUNCTION NUMBER
E94FCTV  DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
E94RET   DS    0A              4   GENERAL RETURN CODE
E94SRET  DS    0AL2            4   SUB RETURN CODE
E94SR2   DS    AL1             4   SUB RETURN CODE 2
E94SR1   DS    AL1             5   SUB RETURN CODE 1
E94MRET  DS    0AL2            6   MAIN RETURN CODE
E94MR2   DS    AL1             6   MAIN RETURN CODE 2
E94MR1   DS    AL1             7   MAIN RETURN CODE 1
E94FHL   EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*
*   EQUATES FOR EXIT RETURN INFORMATION IN &P.RET
*
E94EXNOK EQU   1                    EXIT NOK
E94EXOK  EQU   0                    EXIT OK
E94INFOR DC    X'00'               INFORMATION GIVEN TO EXIT
E94CLTSV EQU   C'0'                TRANS. CLIENT RESSOURCE TO SERV
E94SVTCL EQU   C'1'                TRANS. SERVER RESSOURCE TO CLI.
E94CALCL EQU   C'2'                TRANS. CALL ON CLIENT
E94UNUS  DC    XL3'00'             ****   UNUSED   ****
E94SCBAD DC    A(0)                ADDRESS OF SCB
         DS    CL20                RESERVED FOR EXTENTION
E94PLLEN EQU   *-E94EXBEG          LENGTH OF THE PARAMETER LIST
```

SPOOL exit 095 (output routine)

This SPOOL exit allows systems support to create a user routine which is invoked with every SPOOL output operation. Only during error handling is it not called. The exit routine is called after a channel program has been generated and before this channel program is sent to the printer. The exit routine can be used with line printers and laser printers. APA printers, RSO, remote batch processing, magnetic tapes and floppy disks are not supported.

The exit routine allows you to

– accept the channel program
– modify the channel program
– suppress the channel program

The exit routine is provided with the following information:
R1 = A (EX095 parameter area)

The return codes are entered in the standard header.

Standard header:

dd	cc	bb	aa

A return code concerning execution of EX095 is passed in the standard header.
aa=maincode1; bb=maincode2; cc=subcode1; dd=subcode2

X'aa'	Meaning
00	Channel program output
04	Channel program not output
0C	Channel program not output and EOF

Notes

– The exit routine can access the SPOOL data structures. These data structures are version-specific.

– If the channel program is modified or is not output, the SPOOL accounting record is not modified.

– No code translation is performed on the data passed by the exit routine.

– If the channel program is modified, attention should be paid to the following rules:
  – The channel program and the data buffer must not be extended.
  – If the data addressed by a channel command is shortened, the data length information of the channel command must be corrected accordingly.
  – If the channel program is shortened, the “end-of-chain indicator” must be corrected accordingly.

  - The data address can be modified but it must be an address in the SPOOL output range.

- The “start of output range” address is contained in the first channel command in the chain. The “end of output range” is computed by adding the data length to the data address of the last channel command.

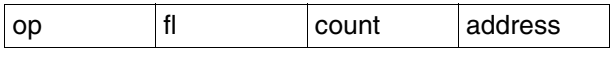
DSECT

A DSECT/CSECT for the parameter area can be generated by means of the EX095 macro. ”prefix” must not be longer than 3 characters.

```
         EX095 D
RECEXPL  DSECT ,                        I/O    EXIT ROUTINE P/L
*
*   STANDARD HEADER RETURN INFORMATION
*
SPOEXBEG DS    0H
         FHDR  MF=(C,SPO),EQUATES=NO
         DS    0A
SPOFHE   DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
SPOIFID  DS    0A              0   INTERFACE IDENTIFIER
SPOFCTU  DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
SPOFCT   DS    AL1             2   FUNCTION NUMBER
SPOFCTV  DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
SPORET   DS    0A              4   GENERAL RETURN CODE
SPOSRET  DS    0AL2            4   SUB RETURN CODE
SPOSR2   DS    AL1             4   SUB RETURN CODE 2
SPOSR1   DS    AL1             5   SUB RETURN CODE 1
SPOMRET  DS    0AL2            6   MAIN RETURN CODE
SPOMR2   DS    AL1             6   MAIN RETURN CODE 2
SPOMR1   DS    AL1             7   MAIN RETURN CODE 1
SPOFHL   EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
*
*   EQUATES FOR EXIT RETURN INFORMATION IN &P.MR1
*
SPOEXIO  EQU   0                    SEND I/O REQUESTED
SPOEXNI  EQU   4                    DO NOT SEND I/O
SPOEXEO  EQU   12                   NO I/O DECLARE EOF
SPOPRTYP DC    AL1(0)              PRINTER TYPE
SPOLINES EQU   32                   LINES PRINTER TYPE          751
```

```
SPOPAGES EQU   33                      PAGE PRINTER TYPE            751
SPOLOADL EQU   34                      LOADABLE LINE PRINTER TYPE   751
SPOINFOR DC    X'00'                   INFORMATION GIVEN TO EXIT
SPONOSCB EQU   X'01'                   IND. SPOSCBAD NOT FILLED
SPOUNUS  DC    XL2'00'                 ****   UNUSED   ****
SPOCHADD DC    A(0)                    CCW'S CHAIN OR BUFFER ADDRESS
SPOSCBAD DC    A(0)                    ADDRESS OF SCB
SPODVTYP DS    AL1                     DEVICE TYPE
SPOPRTMN DS    CL2                     PRINTER MNEMONIC
         DS    CL20                    RESERVED FOR EXTENTION
SPOPLLEN EQU   *-SPOEXBEG              LENGTH OF THE PARAMETER LIST
```

The SPOCHADD field contains the address of the channel program. The format of a channel command is not dependent on the channel or device type.



op = operation code
fl = flag byte
count = data length
address = virtual data address

Notes

– The operation codes of the channel command are dependent on the channel type and printer type and are not checked by SPOOL on reentry from the exit routine.

– The SPONOSCB field indicates that the SPOSCBAD field does not contain an SCB address and cannot therefore be used. This can occur if SPOOL carries out I/O operations which are not assigned to jobs.

– The SPODVTYP field contains the same detailed device type as used in the SPOOL parameter file. The SPSDTYP macro must be used to resolve it.

– The SPOPRTMN field contains the mnemonic device name of the printer.

SPOOL exit 096 (server selection)

This system exit enables SPOOL administration (TSOS or PRINT-SERVICE-ADM) to select specific servers by means of combining the appropriate selection criteria. The selection may be tied to an individual decision table stipulating, for example, that the jobs of user A with paper format A are to be directed to server X, while the jobs of user A with paper format B are to go to server B, and so on.

Exit 096 must be started on the client.

When changes are made to the server list, care should be taken that consistency is not impaired. The server selected by the SPOOL administrator must be contained in the server list passed to the exit. The SPOOL administrator can ensure that this the case by:

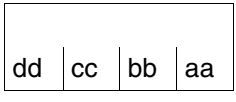
– placing the requested server at the first position in the list and keeping all the other servers

– placing the requested server at the first position in the list and setting the number of available servers in the list to 1.

The following information is passed to the exit routine:
`R1 = A (EX096 parameter area)`

The return codes are entered in the standard header.

Standard header:



A return code concerning execution of EX096 is passed in the standard header.
aa=maincode1; bb=maincode2; cc=subcode1; dd=subcode2

X'aa'	**Meaning**
00	Exit processed without errors
01	Exit processed without errors

DSECT

A DSECT/CSECT for the parameter area can be created using the EX096 macro. “prefix” must not be longer than 3 characters.

```
         EX096 D
SVSEXPL  DSECT ,              SERVER SELECTION EXIT P/L
E96EXBEG DS    0H
         FHDR  MF=(C,E96),EQUATES=NO
         DS    0A
E96FHE   DS    0XL8             0   GENERAL PARAMETER AREA HEADER
*
E96IFID  DS    0A               0   INTERFACE IDENTIFIER
E96FCTU  DS    AL2              0   FUNCTION UNIT NUMBER
*                                   BIT 15    HEADER FLAG BIT,
*                                   MUST BE RESET UNTIL FURTHER NOTICE
*                                   BIT 14-12 UNUSED, MUST BE RESET
*                                   BIT 11-0  REAL FUNCTION UNIT NUMBER
E96FCT   DS    AL1              2   FUNCTION NUMBER
E96FCTV  DS    AL1              3   FUNCTION INTERFACE VERSION NUMBER
*
E96RET   DS    0A               4   GENERAL RETURN CODE
E96SRET  DS    0AL2             4   SUB RETURN CODE
E96SR2   DS    AL1              4   SUB RETURN CODE 2
E96SR1   DS    AL1              5   SUB RETURN CODE 1
E96MRET  DS    0AL2             6   MAIN RETURN CODE
E96MR2   DS    AL1              6   MAIN RETURN CODE 2
E96MR1   DS    AL1              7   MAIN RETURN CODE 1
E96FHL   EQU   8                8   GENERAL OPERAND LIST HEADER LENGTH
*
*
*   EQUATES FOR EXIT RETURN INFORMATION IN &P.RET
*
E96EXNOK EQU   1                     EXIT NOK
E96EXOK  EQU   0                     EXIT OK
E96SCBAD DC    A(0)                 ADDRESS OF SCB
E96SVLAD DC    A(0)                 ADDRESS OF SERVER LIST
         DS    CL12                 RESERVED FOR EXTENTION
E96PLLEN EQU   *-E96EXBEG           LENGTH OF THE PARAMETER LIST
         SPACE 3
E96SVL   DSECT ,              SERVER LIST DESCRIPTION
E96SV#   DS    F                    SERVERS#
E96SVSZ  DS    F                    SIZE
E96SV    DS    0XL20                SERVER(SERVERS#) (1 ENTRY)
E96SVN   DS    CL8                  NAME
E96APR#  DS    FL4                  PRINTERS#
E96DT    DS    XL8                  DEVICE_TYPES
```

SPOOL exit 097 (file transfer control)

SPOOL exit 097 can be used in a client/server environment in conjunction with the product Dprint (Distributed Print Services) to control the number of Dprint jobs for file transfer. Using the exit enables each client to define a maximum number of jobs for file transfer. This causes the number of Dprint system tasks which control the Dprint jobs for which file transfer processing was initiated to be reduced. Furthermore, the exit can be used to prevent the file transfer resources on the client and server processors from being accessed exclusively by Dprint. This in turn may help solve the problem of print jobs being stuck for too long in an intermediate status (WFT or FT).

SPOOL background

If a Dprint job issued by a client has been accepted by a remote server, this job is in the SPOOL job queue of the server (status WFT, wait file transfer). In response to a request from the Dprint file transfer unit, SPOOL selects a job from the queue and forwards it to Dprint. The job selected is the one with the highest priority and whose original host (i.e. the client host which issued the job) is available and has free resources. If the Dprint file transfer unit has received the job, it informs the client host, which then initiates file transfer of the file to be printed and, if appropriate, of the file containing the resources necessary for printing. The Dprint file transfer unit requests jobs without interruption. As a result, all jobs accepted by the server which require file transfer are forwarded for file transfer immediately. If the number of jobs involved is extremely large and the server cannot cope with them all, performance problems will arise on the server side and it will be necessary to make exclusive use of the file transfer resources on the client side. In SPOOL, exit 097 is localized on the server side and comes into play whenever a job is selected for the Dprint file transfer unit.

The following information is transferred to the exit routine:
`R1 = A (EX096 parameter area)`

The following fields are set in the parameter area:

E97HSTI@	Address of a host table containing the names of the available hosts that are not yet saturated
E97HST#	Number of entries in the host table
E97HSTL	Length of an entry in the host table
E97SVT@	Address of the SPOOL vector table

The return code is stored in the standard header of the parameter area.

E97EXNOK	X'00000001': an error occurred during processing.
E97EXOK	X'00000000': the exit was executed without errors.

The following fields can be set by the exit routine:

E97HOK#	Number of hosts left in the input host list *
E97WFT#	Number of jobs with the status WFT for the hosts left in the input host list
E97FT#	Number of jobs with the status FT for the hosts left in the input host list

* input host list = input list from which certain entries have been removed, i.e. replaced by binary zeroes

DSECT

A DSECT/CSECT for the parameter area can be generated by means of the EX097 macro. “prefix” must not be longer than 3 characters.

```
         EX097 D
CLREXPL  DSECT ,           File transfer balancing exit p/l
E97EXBEG DS    0H
         FHDR  MF=(C,E97),EQUATES=NO
         DS    0A
E97FHE   DS    0XL8             0   GENERAL PARAMETER AREA HEADER
*
E97IFID  DS    0A               0   INTERFACE IDENTIFIER
E97FCTU  DS    AL2              0   FUNCTION UNIT NUMBER
*                                   BIT 15    HEADER FLAG BIT,
*                                   MUST BE RESET UNTIL FURTHER NOTICE
*                                   BIT 14-12 UNUSED, MUST BE RESET
*                                   BIT 11-0  REAL FUNCTION UNIT NUMBER
E97FCT   DS    AL1              2   FUNCTION NUMBER
E97FCTV  DS    AL1              3   FUNCTION INTERFACE VERSION NUMBER
*
E97RET   DS    0A               4   GENERAL RETURN CODE
E97SRET  DS    0AL2             4   SUB RETURN CODE
E97SR2   DS    AL1              4   SUB RETURN CODE 2
E97SR1   DS    AL1              5   SUB RETURN CODE 1
E97MRET  DS    0AL2             6   MAIN RETURN CODE
E97MR2   DS    AL1              6   MAIN RETURN CODE 2
E97MR1   DS    AL1              7   MAIN RETURN CODE 1
E97FHL   EQU   8                8   GENERAL OPERAND LIST HEADER LENGTH
*
*
*   EQUATES FOR EXIT RETURN INFORMATION IN &P.RET
*
E97EXNOK EQU   1                     EXIT NOK
E97EXOK  EQU   0                     EXIT OK
E97UNUS  DC    XL4'00'              ****   UNUSED   ****
*
E97HSTL@ DS    A                    ACCESSIBLE HOST LIST
E97SVT@  DS    A                    SVT ADDRESS
E97HST#  DS    H                    NBR OF ENTRIES IN THE LIST
E97HSTL  DS    H                    LENGTH OF AN ENTRY IN THE LIST
*
E97HOK#  DS    H                    OUTPUT AREA = NBR OF CORRECT HST
E97WFT#  DS    H                    TOTAL NBR OF JOB IN WFT
E97FT#   DS    H                    TOTAL NBR OF JOB IN FT
         DS    CL20                 RESERVED FOR EXTENSION
E97PLLEN EQU   *-E97EXBEG           LENGTH OF THE PARAMETER LIST
```

SAT exit 110

The SAT exit routine is only called for events that are to be logged, see the “SECOS“ manual [13]. This only happens immediately before the event is written to the SAT log file. The exit routine must be loaded by systems support and activated by a security administrator (`/MODIFY-SAT-PRESELECTION EXIT=YES`).

The SAT exit routine can then trigger one of the following:
– initiate selective responses (for example blocking a user ID after a certain number of failed LOGON attempts)
– write a separate audit record (ANY event, $SATANY macro)
– on return to SAT via the return code, allow or suppress writing of the analyzed audit record.

The following information is passed to the routine:

```
R1    = A(EX110 parameter area)
R12   = A(TPR program manager)
R13   = A(save area)
R14   = A(indirect return)
R15   = A(exit routine)
```

The return code for the calling system component is sent via the SATZPMDL parameter area in the SATZSR1 field.

SATZSR1 = X'00' The event is to be written to the SAT logging file.

SATZSR1 = X'04' The event is not to be written to the SAT logging file.

The parameter list transferred to the exit routine contains a copy of the audit record. This ensures that the original information cannot be modified.

The parameter list contains the following:
– standard header
– length of the record (2 bytes)
– reserved (2 bytes)
– audit record (fixed and variable parts)

The audit record is described in the “SECOS” manual [13]. It consists of two parts, one fixed and one variable. The variable part contains a list of information fields. There are two types of information field:

The standard information field
– length specification
– identifier (see the “SECOS” manual [13])
– associated data field

The *LNG field
– field with the value 255 (indicator for the *LNG field)
– negative indicator
– length specification
– 2 reserved bytes
– associated data field

The description of the parameter area specifies only the layout for **one** information field (standard information field or *LNG field) of the variable part.

DSECT

```
         EX110 MF=D
         MFTST MF=D,PREFIX=S,MACID=ATZ,ALIGN=F,                          C
               DMACID=ATZ,SUPPORT=(D,C),DNAME=ATZPMDL
SATZPMDL DSECT ,
               *,##### PREFIX=S, MACID=ATZ #####
*   set for subcode_1
SATZLOG  EQU   0                        record-log requested
SATZRJ   EQU   4                        reject record
*
SATZIMDL EQU   5900                     max. length of logging record
*
SATZILFP EQU   28                       length of fixed part in
*                                       logging record
*
*   parameter list description
SATZHDR  FHDR  MF=(C,SATZ),EQUATES=NO                  standard header
SATZHDR  DS    0A
SATZFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
SATZIFID DS    0A              0   INTERFACE IDENTIFIER
SATZFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
SATZFCT  DS    AL1             2   FUNCTION NUMBER
SATZFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
SATZRET  DS    0A              4   GENERAL RETURN CODE
SATZSRET DS    0AL2            4   SUB RETURN CODE
SATZSR2  DS    AL1             4   SUB RETURN CODE 2
SATZSR1  DS    AL1             5   SUB RETURN CODE 1
SATZMRET DS    0AL2            6   MAIN RETURN CODE
SATZMR2  DS    AL1             6   MAIN RETURN CODE 2
SATZMR1  DS    AL1             7   MAIN RETURN CODE 1
```

```
SATZFHL  EQU   8              8   GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
SATZOK   EQU   0                       no error
*
SATZLEN  DS    H                       length of logging record
SATZRS1  DS    XL2                     alignment
*
SATZREC  DS    0XL5928                 logging record
*
SATZFIX  DS    0XL28                   fixed part of logging record
SATZUID  DS    CL8                     user id
SATZTSN  DS    CL4                     TSN
SATZEVT  DS    CL3                     event
SATZRES  DS    CL1                     event result
*
SATZTSP  DS    0XL8                    time stamp
SATZDAT  DS    CL4                     YYYYMMDD
SATZTIM  DS    CL4                     HHMMSS00
*
SATZSAT  DS    X                       SAT version
SATZETT  DS    X                       reserved for extent
SATZSCF  DS    X                       caller
SATZSCV  DS    X                       caller version
*
SATZVAR  DS    0XL5900                 variable part of logging
*                                      record
SATZDAR  DS    CL5900                  variable part
         ORG   SATZVAR
*
SATZFLD  DS    0XL258                  layout of a standard field
SATZVLN  DS    X                       length of the field
SATZVID  DS    CL2                     id of the field
SATZVDT  DS    CL255                   data
*
         ORG   SATZVAR
*
SATZEFLD DS    0XL4007                 layout of a *LNG field
SATZVLND DS    X                       255 to indicate *LNG-field
SATZNIDE DS    CL2                     negative id of the field
SATZFLEN DS    CL2                     length of *LNG-field
SATZABRL DS    CL2                     on exit 110 always zero
SATZLVAL DS    CL4000                  data
*
         ORG   SATZVAR+5900
*
SATZ#    EQU   *-SATZHDR
```

$SATANY macro

Application: system security macro (TPR)
Macro type: GC type (C form, D form, E form, L form, M form)

SAT can be called from a system exit. The $SATANY interface allows privileged users to write data of their own into the SAT logging file. If $SATANY is used in exit routine 110, a protection mechanism prevents recursive exit routine calls. The ANY event can be rejected if the security administrator has not selected it for auditing.

The $SATANY macro is used in privileged mode to write audit records for security-relevant, user-specific events into the SAT logging file.

This macro introduces the global event ANY. A user-specific subcode four characters in length can be created for this event. This option facilitates editing of the events using the evaluator SATUT (cf. the “SECOS” manual [13]).

The user can have the following information audited:

– type of event
– result of event
– sub-event code
– data type
– data reference
– long data reference

SAT complements this information with the following data:

– TSN of the calling task
– user ID of the calling task
– time stamp
– group ID (if provided)
– chipcard ID, if a chipcard is used
  or alternatively
  the personal user ID if SRPM is being used and access via personal identification has been specified for a user ID.

Note

In some cases, particularly in the event of an unsuccessful logon attempt, SAT cannot audit any user IDs. This interface does not permit the calling task’s user ID to be overwritten.

Macro format and operands description

[name] $SATANY
[,TYPE=*NONE/<variable>/*ANY]
[,RESULT=*NONE/<variable>/*SUCC/*FAIL]
[,SUBCOD=*NONE/<variable>/*TEXT/*HEXA/*BOTH]
[,DATATYP=*NONE/<variable>]
[,DATAPTR=*NONE/<variable>]
[,DATALEN=*NONE/<variable>]
[,LDTAPTR=*NONE/<variable>]
[,LDTALEN=*NONE/<variable>]

TYPE=
Determines the event type.

=*NONE
The operand is not used (default with MF=L).

= <variable>
Symbolic address of a three-byte field containing the event type (always ANY).

=*ANY
Event of the type ANY.

RESULT
Defines the result of the event.

=*NONE
The operand is not used (default with MF=L).

=<variable>
Symbolic address of a one-byte field with the event result (SUCC or FAIL).

=*SUCC
The event has been executed completely and successfully.

=*FAIL
A fatal error occurred during the event.

SUBCOD
Subcode for an event.
This determines the name of the subevent within the event. The field must be left-justified. It can be less than or equal to 4 characters. If the subevent is less than 4 characters, the field must be padded with blanks.

=*NONE
The operand is not used (default with MF=L).

=<variable>
Determines the symbolic address of a four-byte field.

DATATYP
Determines the type of information to be audited.

=*NONE
The operand is not used (default with MF=L).

=<variable>
Determines the symbolic address of a two-byte field containing the type of the information to be audited:
X'0060' = type *TEXT
X'0061' = type *HEXA
X'0062' = type *BOTH

=*TEXT
The field is output as a character string.

=*HEXA
The field is output in hexadecimal form.

=*BOTH
Specifies that the field contains both text and hexadecimal characters. The first line of the output contains text, and the next two lines contain the corresponding hexadecimal code (in accordance with the EDT output in hexadecimal mode).

DATAPTR
Data pointer (data must not be longer than 255 bytes).

=*NONE
The operand is not used (default with MF=L).

=<variable>
Symbolic address of a four-byte field containing the address of the first byte of the data field to be audited.

DATALEN
Length of the information.

=*NONE
The operand is not used (default with MF=L).

=<variable>
Symbolic address of a one-byte field containing the size (in bytes) of the data field to be audited.

LDTAPTR
Pointer to data with a length greater than 255 bytes.

=*NONE
The operand is not used (default with MF=L).

=<variable>
Symbolic address of a four-byte field containing the address of the first byte of the data field to be audited.

LDTALEN
Length of the information.

=*NONE
The operand is not used (default with MF=L).

=<variable>
Symbolic address of a two-byte field containing the size (in bytes) of the data field specified by LDTAPTR.

Relationships between the operands:

The following table shows how the $SATANY operands are interconnected, i.e. the mandatory and the optional parameters for the ANY event.

	SAT information							
Event	TYPE	RESULT	SUBCODE	DATATYP	DATAPTR	DATALEN	LDTAPTR	LDTALEN
ANY event	ANY	M	O	O	O	O	O	O

M = mandatory, O = optional

If either DATATYP, DATAPTR or DATALEN is assigned the value *NONE, no data field is audited.
The data field specified by LDTAPTR and LDTALEN is audited only if the security administrator has activated the auditing of additional information for events (`/MODIFY-SAT-PRESELECTION LOGGING-QUANTITY=*EXTENDED`).

Entry name: SATANY

4 Eventing in exit routines

In nonprivileged (TU) user programs the programmer uses the BS2000/OSD bourse mechanism with the ENAEI, SOLSIG, POSSIG... macros (see the “Executive Macros” manual [10]) to control programs by means of eventing.

These macros cannot be used in the privileged (TPR) exit routines.

To make these functions accessible to the exit programmers the three privileged macros described below are provided. They connect:

– an auxiliary task under the system administrator ID TSOS, which creates an event item (with ENAIE)
– via exit routine number 0, which is called within the auxiliary task with EXCALL
– to “normal” exit routines which want to use the mechanism.

These functions enable, for example, several exit routines to write to the same logging file during a system run. To this end an event-driven program is started under the ID TSOS; this program becomes effective every time an exit routine connected to the same event item requires an output.

$GTBOID - get bourse identifier

The $GTBOID macro ascertains the bourse ID associated with an event item. $GTBOID can only be used in exit routine number 0, which is called by the auxiliary task under TSOS. The task also created the event item (ENAEI macro). The event item ID must be transferred to the exit routine in the EXCALL parameter list.

macro format and operands description

[label] $GTBOID
MF=C / D / E / L [,PREFIX=prefix] [,PARAM=addr]

MF — Defines the macro format.
- =C — The parameter list is generated as a data area.
- =D — The parameter list is generated as a DSECT with the name specified for “label”; default value: label = NTEG.
- =E — The commands are generated.
- =L — The parameter list is generated (without field names and equates).

PREFIX — Character for the field names in the parameter list. Only possible in conjunction with MF=C/D.
- =prefix — Character with which the field names in the parameter list are to begin; default value: N.

PARAM — Address of the parameter list; only practical in conjunction with MF=E.
- =addr — Address of the parameter list.

Notes

– R12 and R13 must not have been modified since the exit routine was called.
– R14, R15 and R1 are updated by MF=E.
– The parameter list must be processed in line with reentrant programming.

Return code

The following return code is transferred in the parameter list’s standard header:

Main code	Meaning
NTEGPAER	Parameter syntax error
NTEGEIER	Incorrect event item ID; no bourse ID returned
X' 00'	Successful call; bourse ID returned

```
EXIT0000 @ENTR TYP=E,ENV=SPLSPEC,LOCAL=ADF
          :
         MVC   CPARAM(NTEG#),LPARAM ──────────────────────────────────  (1)
         MVC   NTEGEIID(4),EIID ──────────────────────────────────────  (2)
         $GTBOID MF=E,PARAM=CPARAM ───────────────────────────────────  (3)
         CLI   NTEGMR1,X'00' ─────────────────────────────────────────  (4)
         BNE   exception
         MVC   BOID(4),NTEGBOID ──────────────────────────────────────  (5)
          :
         @EXIT
          :
LPARAM   $GTBOID MF=L ──────────────────────────────────────────────────  (6)
NTEGDATA DS    0A
NTEGFHE  DS    0XL8           GENERAL OPERAND LIST HEADER
NTEGFCTU DC    AL2(81)        FUNCTION UNIT NUMBER
NTEGFCT  DC    AL1(0)         FUNCTION NUMBER
NTEGFCTV DC    AL1(1)         FUNCTION INTERFACE VERSION NUMBER
NTEGRET  DC    X'FFFFFFFF'    RETURN CODE NOT VALID
NTEGMR1  EQU   NTEGRET+3,1    MAIN RETURN CODE 1
*
NTEGEIID DC    F'0'           Event Item Identifier (EIID)
NTEGBOID DC    F'0'           Bourse Identifier (BOID)
NTEG#    EQU   *-NTEGDATA     Size of Parameter List
          :
BOID     DC    A(0)
         ENTRY BOID
         @END
ADF      @PAR  L=YES
          :
CPARAM   $GTBOID MF=C ──────────────────────────────────────────────────  (7)
ADF      @PAR  LEND=YES
```

(1) The current parameter list is preset with the default values.

(2) The event item ID is added.

(3) The function $GTBOID is called.

(4) The return code is evaluated.

(5) The bourse ID is stored in a clearly visible place or in the system vector if the operation is successful.

(6) Standard values for the parameter list.

(7) Dummy character of the current parameter list in automatic memory layout.

$NBSSIG - send signal to event item bourse

The $NBSSIG function enables the exit routine to indicate to a waiting task that a bourse event has occurred (like POSSIG in TU). A post code is transferred to the task waiting for the event with the aid of SOLSIG.
Prerequisite: an event item must exist and the associated bourse ID must be stored with $GTBOID with system-global accessibility.

macro format and operands description

[label] $NBSSIG
MF=C / D / E / L / M [,PREFIX=prefix] [,PARAM=addr] [,BOID=boid / (R1)] [,SDATA=(1, addr / (R2))]

MF	Defines the macro format.
=C	The parameter list is generated as a data area.
=D	The parameter list is generated as a DSECT with the name specified for “label”; default value: label = NTEG.
=E	Function call.
=L	The parameter list is generated (without field names and equates).
=M	Modification of the parameter list with current values.
PREFIX	Character for the field names in the parameter list. Only possible in conjunction with MF=C/D.
=prefix	Character with which the field names in the parameter list are to begin; default value: N.
PARAM	Address of the parameter list if MF=E is specified.
=addr	Address of the parameter list.
BOID=	Bourse ID.
=boid	Bourse ID stored globally using $GTBOID.
=(R1)	Register containing the bourse ID.
SDATA	Field with the address of the 1-word post code transferred to the waiting task with SOLSIG. This can be, for example, the address of the data to be output into an audit file.
=(1,...)	addr = address of the field containing the 1-word post code. R2 = register with the address value of the field containing the 1-word post code.

Notes

– R12 and R13 must not have been modified since the exit routine was called.

– R14, R15 and R1 are updated by MF=E.

– The parameter list must be processed in line with reentrant programming.

Return codes

The following return codes are transferred in the parameter list's standard header (using $NDSSIG MF=D/C):

Maincode	Meaning
NTBSM1OK	Successful call
NTBSM1IC	Inconsistent parameters
NTBSM1BE	Error in bourse data
NTBSM1IB	Invalid bourse data
NTBSM1BD	Bourse already destroyed, i.e. event item no longer exists
NTBSM1QL	Overflow in the bourse send queue
NTBSM1NS	No slots available for the send queue

No action is required for any of the return codes except NTBSM1OK.

Programming example

```
EXITXXXX @ENTR TYP=E,ENV=SPLSPEC,LOCAL=ADF
          :
         L     R1,BOIDX ------------------------------------------------  (1)
         L     R1,0(R1)
         MVC   CPARAM(NTBS#),LPARAM ------------------------------------  (2)
         $NBSSIG MF=M,PARAM=CPARAM,BOID=(R1),SDATA=(1,SDATA) ----------  (3)
         $NBSSIG MF=E,PARAM=CPARAM ------------------------------------  (4)
         CLI   NTBSMR1,NTBSM1OK ---------------------------------------  (5)
         BNE   exception
          :
         @EXIT
          :
LPARAM   $NBSSIG MF=L ------------------------------------------------  (6)
NTBSS#   DS    0A
NTBSFHE  DS    0XL8           GENERAL OPERAND LIST HEADER
NTBSFCTU DC    AL2(93)        FUNCTION UNIT NUMBER
NTBSFCT  DC    AL1(6)         FUNCTION NUMBER
NTBSFCTV DC    AL1(1)         FUNCTION INTERFACE VERSION NUMBER
NTBSRET  DC    X'FFFFFFFF'    RETURN CODE NOT VALID
NTBSMR1  EQU   NTBSRET+3,1    MAIN RETURN CODE 1
*
```

```
NTBSBOID DC    F'0'           BOID
         DC    X'00'          unused
         DC    X'00'          unused
NTBSVECT DC    B'01000000'    VECT. FIELD
NTBSDLTH DC    X'00'          DATA LENGTH
NTBSLTIM DC    H'600'         LIFETIME
NTBSPRIO DC    AL1(0)         PRIORITY
         DC    X'00'          unused
NTBSSIG  DC    F'0'           SIGNAL
         DC    F'0'           unused
         DC    F'0'           unused
NTBSSDTA DC    A(0)           SEND DATA ADDRESS
NTBSARSL DC    A(0)           AR Slot address (only for SIH user)
NTBS#    EQU   *-NTBSS#       Length of parameterlist
          :
BOIDX    DC    A(BOID) ------------------------------------------------  (7)
         EXTRN BOID
SDATA    DC    A(MESS)
MESS     DC    Y(MESSEND-MESS)
         DC    C'  ANY KIND OF TEXT'
         :
         @END
ADF      @PAR  L=YES
          :
CPARAM   $NBSSIG MF=C ------------------------------------------------  (8)
          :
ADF      @PAR  LEND=YES
```

(1) The bourse ID is supplied with a value.

(2) The current parameter list is preset with the default values.

(3) The bourse ID and the post code are added.

(4) The function $NBSSIG is called.

(5) The return code is evaluated.

(6) Standard values for the parameter list.

(7) Example of how the bourse ID is determined and the post code used.

(8) Dummy character of the current parameter list in automatic memory layout.

5 System exits for program libraries

With the aid of the system exits for program libraries (PLAM exits) systems support can monitor (and if necessary deny) access to the library members. The following functions are supported: open, close, delete and rename member. Each of these functions has a request and a return exit, i.e. before calling the function and after the operation has been carried out (cf. the table of PLAM exits below).

Selection is made via a separate exit, which is activated whenever the ATTACH library function is called. The exits for the member access functions can thus be limited to an acceptable minimum (performance).

Function	**Type of exit**	**Exit number**
Open member (OPEN ELEMENT)	request return	005 006
Close member (CLOSE ELEMENT)	request return	007 008
Delete member (DELETE ELEMENT)	request return	055 056
Rename member (RENAME ELEMENT)	request return	057 058
Open program library (ATTACH LIBRARY)		054

The following information is passed to the exit points:

```
R1    = A(exit parameter area)
R12   = A(TPR program manager)
R13   = A(save area of the calling component)
R14   = A(indirect return)
R15   = A(exit routine)
```

Return code:

```
R15 =    X'000000RR'

         RR =   Return code of exit routine
                00  no error (005-008, 055-058)
                    library to be monitored (054)
                04  function is rejected (005, 006, 008, 055-058)
                    library not to be monitored (054)
```

At the request exit systems support can reject the function call before it is performed (exit points 005, 055, 057).

At the return exit the function has already been performed (exit points 006, 008, 056, 058). With the exception of exit point 006 (OPEN ELEMENT) the action can no longer be revoked.

DSECT

All the exits for program libraries are provided with a parameter list with the same structure.

The DSECT for the parameter list can be generated by means of the EXnnn macro:

```
name  EXnnn   D[,[prefix],[ALL]]
```

nnn = 3-digit exit number

If ALL is not specified, only the fields supplied with values by the function are provided with names. If ALL is specified, a parameter list valid for all functions is generated. The parameter “prefix” enables the symbolic field names to be prefixed by a freely selectable alphabetic character.

All exits in the parameter list are provided with the current PLAM version and library name. If specified previously, LINK=name is stored. The type, name, version and variant of a library member are also stored in the parameter list. This is not true for:

– ATTACH LIBRARY - there is no connection to the library member as yet.

– OPEN ELEMENT - in the case of the request exit, the specifications for the library member may still be incomplete (see OPEN member exit 005).

The following table illustrates which fields have values supplied by PLAM for the different exit routines.

Fields supplied by PLAM	Exit routine								
	ATTACH	OPEN		CLOSE		DELETE		RENAME	
	054	005	006	007	008	055	056	057	058
PMRC	x	x	x	x	x	x	x	x	x
PMRC1	x	x	x	x	x	x	x	x	x
PMPLV	x	x	x	x	x	x	x	x	x
PMLIB	x	x	x	x	x	x	x	x	x
PMLINK	x	x	x	x	x	x	x	x	x
PMOPM		x	x						
PMSLV		x	x						
PMLVD		x	x						
PMCLM				x	x				
PMTYP		x	x	x	x	x	x	x	x
PMNAM		x	x	x	x	x	x	x	x
PMVER		x	x	x	x	x	x	x	x
PMVAR		x	x	x	x	x	x	x	x
PMDAT		x	x	x	x	x	x	x	x
PMTYP2								x	x
PMNAM2								x	x
PMVER2								x	x

6 Appendix

6.1 System tables

This section contains DSECTs for various system tables mentioned in the foregoing text. The macro expansions given here are valid for BS2000/OSD-BC V5.0. For actual use in practice users are recommended to generate the appropriate tables themselves.

```
CATAL MF=D,VERSION=3
*********    CATAL PARAMETER AREA  --  VERSION=3 *******
         MFTST MF=D,PREFIX=I,MACID=DK,ALIGN=F,                          C
               DMACID=DK,SUPPORT=(C,D),DNAME=DKCATPL
IDKCATPL DSECT ,
               *,##### PREFIX=I, MACID=DK #####
IDKCATST EQU   *
         FHDR  MF=(C,IDK),EQUATES=NO
         DS    0A
IDKFHE   DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
IDKIFID  DS    0A              0   INTERFACE IDENTIFIER
IDKFCTU  DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
IDKFCT   DS    AL1             2   FUNCTION NUMBER
IDKFCTV  DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
IDKRET   DS    0A              4   GENERAL RETURN CODE
IDKSRET  DS    0AL2            4   SUB RETURN CODE
IDKSR2   DS    AL1             4   SUB RETURN CODE 2
IDKSR1   DS    AL1             5   SUB RETURN CODE 1
IDKMRET  DS    0AL2            6   MAIN RETURN CODE
IDKMR2   DS    AL1             6   MAIN RETURN CODE 2
IDKMR1   DS    AL1             7   MAIN RETURN CODE 1
IDKFHL   EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
IDKPN1   DS    CL80          FILE                          .#'008' 200
*
IDKPN2   DS    CL80          NEWNAME                       .#'058' 200
*
*                            PASSWORDS
IDKRDPAS DS    F               READ  PASSWORD              .#'0A8'
IDKWRPAS DS    F               WRITE PASSWORD              .#'0AC'
IDKEXPAS DS    F               EXEC  PASSWORD              .#'0B0'
*
IDKRETPD DS    H             RETENTION PERIOD              .#'0B4'
*
IDKCFLAG DS    X             FLAG                          .#'0B6'
IDKSTATE EQU   X'80'           7-7 :S: STATE=*UPDATE
*                                  :R: STATE=*NEW OR STATE=*FOREIGN
IDKACCES EQU   X'40'           6-6 :S: ACCESS=*READ
*                                  :R: ACCESS=*WRITE
IDKSHARE EQU   X'20'           5-5 :S: SHARE=*YES / :R: SHARE = *NO
IDKRPSP  EQU   X'10'           4-4 :S: RDPASS  SPECIFIED          110
IDKWPSP  EQU   X'08'           3-3 :S: WRPASS  SPECIFIED          110
```

```
IDKRPDSP EQU   X'04'           2-2 :S: RETPD   SPECIFIED           110
IDKACCSP EQU   X'02'           1-1 :S: ACCESS  SPECIFIED           110
IDKSHASP EQU   X'01'           0-0 :S: SHARE   SPECIFIED           110
*
IDKCFLG1 DS    X             FLAG1                          .#'0B7'
IDKCRSP  EQU   X'80'           7-7 :S: RELSPAC SPECIFIED           110
IDKCBAS  EQU   X'40'           6-6 :S: BASE   SPECIFIED            953
IDKNOENC EQU   X'20'           5-5 :S: DO NOT ENCRYPT PASSWORDS
IDKSTATF EQU   X'10'           4-4 :S: STATE = *FOREIGN
IDKKEEP  EQU   X'08'           3-3 :S: DISP = *KEEP                953
IDKREUS  EQU   X'04'           2-2 :S: DISP = *REUSE
IDKCYCL  EQU   X'02'           1-1 :S: DISP = *CYCLE
IDKDELT  EQU   X'06'           1-2 :S: DISP = *DELETE
IDKDISPS EQU   X'0E'           1-2-3 : IF NOT ZERO, DISP PRESENT   200
IDKEXSP  EQU   X'01'           0-0 :S: EXPASS SPECIFIED            110
*
*                          GENERATION FIELDS
IDKBASE  DS    H               BASE  = +/-NNN OR NNNN      .#'0B8'
IDKGEN   DS    H               GEN   = NNN                 .#'0BA'
IDKFRST  DS    H               FIRST = NNNN                .#'0BC'
IDKDEVTP DS    CL8             NAME OF DEVICE-TYPE         .#'0BE' 910
IDKVOLN  DS    CL6             VOLUME NAME OF PRIVATE DISK .#'0C6' 004
*
IDKBKUP  DS    X             BACKUP                         .#'0CC' 005
IDKBKUPU EQU   X'8F'           7-3-2-1-0 :R: -- UNUSED, MUST BE 0 --
IDKBKUPE EQU   X'40'           6-6 :S: BACKUP LEVEL = E
IDKBKUPD EQU   X'30'           4-5 :S: BACKUP LEVEL = D
IDKBKUPC EQU   X'20'           5-5 :S: BACKUP LEVEL = C
IDKBKUPB EQU   X'10'           4-4 :S: BACKUP LEVEL = B
IDKBKUPA EQU   X'00'                   BACKUP LEVEL = A
*
IDKCFLG2 DS    X             FLAG2                          .#'0CD' 005
IDKRSPC  EQU   X'80'           7-7 :S: RELSPAC      = *IGNORED     110
IDKSPECI EQU   X'40'           6-6 :S: SHARE        = *SPECIAL
IDKPWSK  EQU   X'20'           5-5 :S: PASSW        = *YES
IDKBKUPF EQU   X'10'           4-4 :S: BACKUP LEVEL   SPECIFIED
IDKDESTU EQU   X'0C'           2-3 :S: DESTROY      = *UNCHANGED   200
IDKNDEST EQU   X'08'           3-3 :S: DESTROY      = *NO
IDKDEST  EQU   X'04'           2-2 :S: DESTROY      = *YES
IDKNLARG EQU   X'02'           1-1 :S: LARGE        = *NO
IDKLARGE EQU   X'01'           0-0 :S: LARGE        = *YES
*
IDKAUDIT DS    X             AUDIT                          .#'0CE'
IDKAUDUN EQU   X'F0'           4-7     -- UNUSED, MUST BE 0 --     200
IDKAUDNO EQU   X'08'           3-3 :S: AUDIT        = *NONE
IDKAUDAL EQU   X'04'           2-2 :S: AUDIT        = *ALL
IDKAUDFA EQU   X'02'           1-1 :S: AUDIT        = *FAILURE
IDKAUDSU EQU   X'01'           0-0 :S: AUDIT        = *SUCCESS
```

```
IDKNOAUD EQU   X'00'                    AUDIT          NOT SPECIFIED
*
IDKFLAG3 DS    X              FLAG3                          .#'0CF'
IDKNERIM EQU   X'80'            7-7 :S: DO NOT ERASE BY IMCAT (TP)  950
IDKSUSA  EQU   X'40'            6-6 :S: NO PUBSPACE ACCOUNTING(TP)  950
IDKOPNBY EQU   X'20'            5-5 :S: OPNBACK=*YES                001
IDKOPNBN EQU   X'10'            4-4 :S: OPNBACK=*NO                 001
IDKACLPY EQU   X'08'            3-3 :S: ACLPROT=*YES          (TP)  005
                                              15:29:06  2001-10-15   PAGE
0030
SOURCE STATEMENT
IDKACLPN EQU   X'04'            2-2 :S: ACLPROT=*NO           (TP)  005
IDKBACLN EQU   X'02'            1-1 :S: BASACL=*NONE                010
IDKBACLS EQU   X'01'            0-0 :S: BASACL=*STD                 014
*
IDKMIGRI DS    X              MIGRATE                        .#'0D0'
IDKMIGAL EQU   X'01'            0-0 :S: MIGRATE = *ALLOWED          951
IDKMIUNU EQU   X'02'            1-1 :R: -- UNUSED --                010
IDKMIGIN EQU   X'03'            0-1 :S: MIGRATE = *INHIBITED        951
IDKMIGFB EQU   X'04'            2-2 :S: MIGRATE = *FORBIDDEN        200
IDKMIGNO EQU   X'00'                :R: MIGRATE NOT SPECIFIED       951
IDKMUNUS EQU   X'F8'            -- UNUSED, MUST BE 0 --             951
*
IDKFAR   DS    0X             ACCESS RIGHTS                       003
IDKOWNER DS    X                OWNER                        .#'0D1' 003
IDKGROUP DS    X                GROUP                        .#'0D2' 003
IDKOTHER DS    X                OTHERS                       .#'0D3' 003
*
IDKFAUS  EQU   X'80'                7-7 :S: USER CLASS SPECIFIED      003
IDKFARS  EQU   X'40'                6-6 :S: READ   SPECIFIED          003
IDKFAWS  EQU   X'20'                5-5 :S: WRITE  SPECIFIED          003
IDKFAXS  EQU   X'10'                4-4 :S: EXEC   SPECIFIED          003
IDKFARO  EQU   X'08'                3-3 :S: SET READ                  003
IDKFAWO  EQU   X'04'                2-2 :S: SET WRITE                 003
IDKFAXO  EQU   X'02'                1-1 :S: SET EXEC                  003
IDKFARU  EQU   X'01'                0-0 -- UNUSED, MUST BE 0 ---      003
*
IDKLAST  DS    H              LAST = NNNN                    .#'0D4' 110
*
IDKPERF  DS    X              IO-PERFORMANCE                 .#'0D6' 110
IDKPFUM  EQU   X'FF'            0-7 :S: IOPERF = *USER_MAX          110
IDKPFVH  EQU   X'03'            0-1 :S: IOPERF = *VERY_HIGH         110
IDKPFHI  EQU   X'02'            1-1 :S: IOPERF = *HIGH              110
IDKPFST  EQU   X'01'            0-0 :S: IOPERF = *STD               110
IDKPFNS  EQU   X'00'                    IOPERF NOT SPECIFIED        110
*
IDKUSAG  DS    X              IO-USAGE                       .#'0D7' 110
IDKUSU   EQU   X'FC'            2-7  -- MUST BE 0, RES. FOR FUTURE USE
```

```
IDKUSRW  EQU   X'03'            0-1 :S: IOUSAGE = *READ_WRITE       110
IDKUSWR  EQU   X'02'            1-1 :S: IOUSAGE = *WRITE            110
IDKUSRD  EQU   X'01'            0-0 :S: IOUSAGE = *READ             110
IDKUSNS  EQU   X'00'                    IOUSAGE NOT SPECIFIED       110
*
IDKDW    DS    X              DISK-WRITE                    .#'0D8' 110
IDKDWU   EQU   X'FC'            2-7  -- MUST BE 0, RES. FOR FUTURE USE
IDKDWCL  EQU   X'02'            1-1 :S: DISKWR = *BY_CLOSE          110
IDKDWIM  EQU   X'01'            0-0 :S: DISKWR = *IMMEDIATE         110
IDKDWNS  EQU   X'00'                    DISKWR NOT SPECIFIED        110
*
IDKFLAG4 DS    X              FLAG 4                        .#'0D9' 110
IDKGRDN  EQU   X'80'            7-7 :S: GUARDS = *NONE              117
IDKFTS   EQU   X'40'            6-6     -- RESERVED FOR FTS --      117
IDKCCSST EQU   X'20'            5-5 :S: CCS = *STD                  117
IDKWRKGP EQU   X'10'            4-4 :S: WORKGRP = *YES              201
IDKRDGS  EQU   X'08'            3-3 :S: READ GUARD SPECIFIED        110
IDKWRGS  EQU   X'04'            2-2 :S: WRITE GUARD SPECIFIED       110
                                              15:29:06  2001-10-15   PAGE
0031
SOURCE STATEMENT
IDKXGS   EQU   X'02'            1-1 :S: EXEC GUARD SPECIFIED        110
IDKCCSS  EQU   X'01'            0-0 :S: CODED CHARACTER SET SPEC.   110
*
IDKRDG   DS    CL18           READ GUARD                    .#'0DA' 110
IDKWRG   DS    CL18           WRITE GUARD                   .#'0EC' 110
IDKXG    DS    CL18           EXEC GUARD                    .#'0FE' 110
*
IDKCHECK DS    X              CHECK                         .#'110' 200
IDKCKSTD EQU   0                *STD (DEFAULT)                      200
IDKCKNO  EQU   1                *NO                                 200
IDKCKERR EQU   2                *ERROR                              200
IDKCKSIN EQU   3                *SINGLE                             200
IDKCKMUL EQU   4                *MULTIPLE                           200
IDKCKUID EQU   5                *USERID                             200
IDKCKCAT EQU   6                *CATALOG                            200
*
IDKLIST  DS    X              LIST                          .#'111' 200
IDKLSNO  EQU   0                *NO (DEFAULT)                       200
IDKLSEOU EQU   1                *ERRORS_TO_SYSOUT                   200
IDKLSOUT EQU   2                *SYSOUT                             200
*
IDKAVAIL DS    X              AVAIL                         .#'112' 200
IDKAVANS EQU   0                NOT SPECIFIED                       200
IDKAVAST EQU   1                *STD                                200
IDKAVAHI EQU   2                *HIGH                               200
*
IDKSOMIG DS    X              SOMIGR                        .#'113' 200
```

```
IDKSOMNS EQU   0               NOT SPECIFIED                          200
IDKSOMAL EQU   1               *ALLOWED                               200
IDKSOMFB EQU   4               *FORBIDDEN                             200
*
IDKCCS   DS    CL8           CODED CHARACTER SET              .#'114' 110
*
IDKFTAM  DS    XL8           RESERVED AREA FOR FTAM           .#'11C' 110
*
IDKFLAG6 DS    X             FLAG 6 (ONLY FOR TPR CALL)       .#'124' 116
IDKFLG6U EQU   X'F8'           4-7 MUST BE 0, RES. FOR FUTURE USE  116
IDKPECTC EQU   X'04'           3-3 :S: RES. FOR DMS                204
IDKPCOPY EQU   X'03'           1-2 :S: RES. FOR DMS                116
IDKPMODI EQU   X'02'           2-2 :S: RES. FOR DMS                116
IDKPCREA EQU   X'01'           1-1 :S: RES. FOR DMS                116
*
IDKTPFG1 DS    X             TRR FLAG1                        .#'125' 120
IDKDNERN EQU   X'80'           7-7 :S: DNTERAS=*NO (TPR)           120
IDKDNERY EQU   X'C0'           6-7 :S: DNTERAS=*YES (TPR)          120
IDKURSFN EQU   X'20'           5-5 :S: USRSYSF=*NO (TPR)           120
IDKURSFY EQU   X'30'           4-5 :S: USRSYSF=*YES (TPR)          120
IDKERF5N EQU   X'08'           3-3 :S: ERADF5R=*NO (TPR)           120
IDKERF5Y EQU   X'0C'           2-3 :S: ERADF5R=*YES (TPR)          120
IDKSRMBN EQU   X'02'           1-1 :S: SRMBKUP=*NO (TPR)           201
IDKSRMBY EQU   X'03'           0-1 :S: SRMBKUP=*YES (TPR)          201
*
IDKUNUS1 DS    X             -- UNUSED --                     .#'126' 200
*
IDKPROTP DS    X             PROTECT - TYPE                   .#'127' 200
IDKPRONS EQU   0               PROTECT NOT SPECIFIED               200
IDKPROFI EQU   1               PROTECT=(*FROM_FILE,...)            200
IDKPROST EQU   2               PROTECT=*STD                        200
*
IDKSECRF DS    CL54          PROTECT - REFERENCE              .#'128' 200
*
IDKPRFLG DS    X             PROTECT - FLAG                   .#'15E' 200
IDKSHAUC EQU   X'80'           7-7 :S: SHARE=*UNCHANGED            200
IDKACCUC EQU   X'40'           6-6 :S: ACCESS=*UNCHANGED           200
IDKBALUC EQU   X'20'           5-5 :S: BASACL=*UNCHANGED           200
IDKGRDUC EQU   X'10'           4-4 :S: GUARDS=*UNCHANGED           200
IDKRSPUC EQU   X'08'           3-3 :S: RELSPAC=*UNCHANGED          200
IDKEXDUC EQU   X'04'           2-2 :S: EXDATE=*UNCHANGED           200
IDKPRFGU EQU   X'03'           0-1     -- UNUSED, MUST BE 0 --     200
*
IDKFLAG7 DS    X             FLAG 7                           .#'15F' 200
IDKWCFBD EQU   X'80'           7-7 :S: WILDCARDS FORBIDDEN         200
IDKEXDSP EQU   X'40'           6-6 :S: EXDATE SPECIFIED            200
IDKDEDSP EQU   X'20'           5-5 :S: DELDATE SPECIFIED           200
IDKMANSP EQU   X'10'           4-4 :S: MANCLAS SPECIFIED           200
```

```
IDKUSISP EQU   X'08'           3-3 :S: USRINFO SPECIFIED          200
IDKADISP EQU   X'04'           2-2 :S: ADMINFO SPECIFIED          200
IDKTBLTI EQU   X'01'           0-0 :S: TIMEBASE LTI               210
IDKFLG7U EQU   X'02'           1-1     -- UNUSED, MUST BE 0 --    210
*
IDKEXDAT DS    CL10          EXDATE DATE                  .#'160' 200
IDKEXTIM DS    CL8           EXDATE TIME                  .#'16A' 210
*
IDKDEDAT DS    CL10          DELDATE DATE                 .#'172' 200
IDKDETIM DS    CL8           DELDATE TIME                 .#'17C' 210
*
IDKMANCL DS    CL8           MANCLAS                      .#'184' 200
*
IDKUSINF DS    CL8           USRINFO                      .#'18C' 200
*
IDKADINF DS    CL8           ADMINFO                      .#'194' 200
*
IDKUNUS2 DS    XL1           -- UNUSED2, MUST BE 0 --     .#'19C' 200
*
IDKSTCLF DS    XL1           STOCLAS                      .#'19D' 200
IDKSCNSP EQU   0               NOT SPECIFIED                    200
IDKSCNON EQU   1               *NONE                            200
IDKSCUPD EQU   2               *UPDATE                          200
IDKSCSTD EQU   3               *STD                             200
IDKSCNAM EQU   4               NAME SPECIFIED                   200
IDKSTCLN DS    CL8           STOCLAS-NAME                 .#'19E' 200
*
IDKFUTUR DS    2XL1          -- UNUSED, MUST BE 0 --      .#'1A6' 210
IDKPLLEN EQU   *-IDKCATST    LENGTH OF PARAMETERAREA = X'1A8' (424)
```

```
         FILE MF=D,VERSION=3
              *,VERSION 400
*-------------------------------------------------------*
*        FILE MACRO PARAMETER LIST VERSION 3            *
*-------------------------------------------------------*
         #INTF INTNAME=FILE,                                          C
                                              15:29:06  2001-10-15   PAGE
0033
SOURCE STATEMENT
               REFTYPE=REQUEST,                                       C
               INTCOMP=4
         MFPRE MF=D,PREFIX=,                                        201C
               DNAME=DBPFLPL,MACID=DB,DMACID=DB,ALIGN=F             201
IDBPFLPL DSECT ,
               *,##### PREFIX=I, MACID=DB #####
IDBPFLST EQU   *
         FHDR  MF=(C,IDBF),EQUATES=NO
         DS    0A
IDBFFHE  DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
IDBFIFID DS    0A              0   INTERFACE IDENTIFIER
IDBFFCTU DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
IDBFFCT  DS    AL1             2   FUNCTION NUMBER
IDBFFCTV DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
IDBFRET  DS    0A              4   GENERAL RETURN CODE
IDBFSRET DS    0AL2            4   SUB RETURN CODE
IDBFSR2  DS    AL1             4   SUB RETURN CODE 2
IDBFSR1  DS    AL1             5   SUB RETURN CODE 1
IDBFMRET DS    0AL2            6   MAIN RETURN CODE
IDBFMR2  DS    AL1             6   MAIN RETURN CODE 2
IDBFMR1  DS    AL1             7   MAIN RETURN CODE 1
IDBFFHL  EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
IDBLINK  DS    CL8                 LINK
IDBFILE  DS    CL54                FILENAME
IDBRETPD DS    Y                   RETENTION PERIOD
IDBRECSI DS    Y                   RECORD SIZE
IDBBLKSZ DS    Y                   BLOCK SIZE                       004
IDBKEYPO DS    Y                   KEY POSITION
IDBKEYLE DS    AL1                 KEY LENGTH
IDBCTRLI DS    X                   BLKCTRL-INDIKATOR                952
IDBCTRLN EQU   X'80'                7-7 S BLKCTRL=NO                 952
IDBCTRLP EQU   X'40'                6-6 S BLKCTRL=PAMKEY             952
```

```
IDBCTRLD EQU   X'20'          5-5 S BLKCTRL=DATA                952
IDBCTRLO EQU   X'10'          4-4 S BLKCTRL=NULL                952
IDBCTRLB EQU   X'08'          3-3 S BLKCTRL=*BY-PROG            200
IDBCTRLR EQU   X'F8'          7-3 R BLKCTRL=NOT SPECIFIED       200
IDBCTRLU EQU   X'04'              R -- RESERVED, MUST BE 0 --   200
IDBCF4K  EQU   X'02'          1-1 S BLOCK CONTROL FIELD 4K      200
IDBCF2K  EQU   X'01'          0-0 S BLOCK CONTROL FIELD 2K      200
IDBPAD   DS    AL1            PAD
IDBOPEN  DS    X              OPEN TYPE
IDBSINOT EQU   X'80'          7-7 S SINOUT
IDBOUTIN EQU   X'40'          6-6 S OUTIN
IDBINOUT EQU   X'20'          5-5 S INOUT
IDBUPDAT EQU   X'10'          4-4 S UPDATE
IDBEXTEN EQU   X'08'          3-3 S EXTEND
IDBOUTPU EQU   X'04'          2-2 S OUTPUT
IDBREVER EQU   X'02'          1-1 S REVERSE
IDBINPUT EQU   X'01'          0-0 S INPUT
IDBDEVL  DS    CL8            DEVICE TYP                        950
IDBIND1  DS    X              INDICATORS (SET,RESET)
IDBLABEL EQU   X'60'          6-5   FILE LABEL OPTIONS
IDBSTD   EQU   X'60'              R STD LABELS
IDBNSTD  EQU   X'40'              S NSTD LABELS
IDBLABNO EQU   X'20'              S LABEL=NO
IDBAMN   EQU   X'18'          4-3   PRINTER CONTROL
IDBA     EQU   X'18'              S ASA CONTROL CHARS
IDBM     EQU   X'08'              S EBCDIC CONTROL CHARS
IDBN     EQU   X'18'              R NO CONTROL CHARS
IDBRECFO EQU   X'06'          2-1   RECORD FORM
IDBF     EQU   X'04'              S FIXED LENGTH
IDBV     EQU   X'02'              S VARIABLE LENGTH
IDBU     EQU   X'06'              S UNDEFINED
IDBTPMRK EQU   X'01'          0-0 S TPMARK=YES
IDBIND1U EQU   X'80'              R -- RESERVED, MUST BE 0 --   950
IDBIND2  DS    X              INDICATORS (SET,RESET)
IDBFCBTY EQU   X'C0'          7-6   FCB TYPE
IDBSAM   EQU   X'C0'              R SAM
IDBISAM  EQU   X'40'              S ISAM
IDBPAM   EQU   X'C0'              S PAM
IDBBTAM  EQU   X'80'              S BTAM
IDBDUPKE EQU   X'20'          5-5 S DUPLICATE KEYS ALLOWED
IDBDUPKO EQU   X'20'          5-5 R DUPEKY=NO
IDBOVERL EQU   X'10'          4-4 S OVERLAP=YES
IDBOVERO EQU   X'10'          4-4 R OVERLAP=NO
IDBSHARW EQU   X'08'          3-3 S SHARUPD=WEAK                050
IDBSHARU EQU   X'04'          2-2 S SHARUPD=YES
*                             3-2 R SHARUPD=NO
IDBSHARN EQU   X'01'          0-0 S SHARUPD SPECIFIED
IDBIND2U EQU   X'02'              R -- RESERVED, MUST BE 0 --   050
```

```
IDBIND3  DS    X              INDICATORS (SET,RESET)
IDBSPAKE EQU   X'80'           7-7 S SPACE=(...,*KEEP)           050
IDBBSSTD EQU   X'40'           6-6 S BLKSIZE=STD                 004
IDB2NDSP EQU   X'20'           5-5 S 2ND SPACE PARAM SPECIFIED   004
IDBSPACE EQU   X'10'           4-4 S SPACE=ABS
IDBSTATE EQU   X'08'           3-3 S STATE=FOREIGN
IDBVOLUM EQU   X'04'           2-2 S VOLUME=PRIVATE
IDBMNTQO EQU   X'02'           1-1 S MOUNT=0
IDBDMYPR EQU   X'01'           0-0 S *DUMMY & VOL=PRIVATE (TPR)  009
IDBIND4  DS    X              INDICATORS (SET,RESET)
IDBFILEN EQU   X'80'           7-7 S FILE NOT SPECIFIED
IDBCLOSN EQU   X'40'           6-6 S CLOSE NOT SPECIFIED         052
IDBBLKSN EQU   X'20'           5-5 S BLKSIZE=NOT SPECIFIED
IDBDUPKN EQU   X'10'           4-4 S DUPEKY=NOT SPECIFIED
IDBRETPN EQU   X'08'           3-3 S RETPD=NOT SPECIFIED
IDBPADN  EQU   X'04'           2-2 S PAD=NOT SPECIFIED
IDBKEYPN EQU   X'02'           1-1 S KEYPOS=NOT SPECIFIED
IDBOPENN EQU   X'01'           0-0 S OPEN=NOT SPECIFIED
IDBIND5  DS    X              INDICATORS (SET,RESET)
IDBKEYLN EQU   X'80'           7-7 S KEYLEN=NOT SPECIFIED
IDBLABEN EQU   X'40'           6-6 S LABEL=NOT SPECIFIED
IDBTVSNT EQU   X'20'           5-5 S TVSN=SPECIFIED
IDBOVERN EQU   X'10'           4-4 S OVERLAP=NOT SPECIFIED
IDBRECSN EQU   X'08'           3-3 S RECSIZE=NOT SPECIFIED
IDBDEVIN EQU   X'04'           2-2 S DEVICE=NOT SPECIFIED
IDBFCBTN EQU   X'01'           0-0 S FCBTYPE=NOT SPECIFIED
IDBIND5U EQU   X'02'               R -- RESERVED, MUST BE 0 --   950
IDBIND6  DS    X              INDICATORS (SET,RESET)
IDBRECFN EQU   X'80'           7-7 S RECFORM=NOT SPECIFIED
IDBVOLUN EQU   X'40'           6-6 S VOLUME=NOT SPECIFIED
IDBSPACN EQU   X'20'           5-5 S SPACE=NOT SPECIFIED
IDBSTATN EQU   X'08'           3-3 S STATE=NOT SPECIFIED
IDBMOUNN EQU   X'02'           1-1 S MOUNT=NOT SPECIFIED
IDBIND6U EQU   X'15'               R -- RESERVED, MUST BE 0 --   950
IDBVOL@  DS    A              ADDRESS OF VOLUME-LIST            004
IDBDVOL@ DS    A              ADDRESS OF DVOLUME-LIST           004
IDBTVSN@ DS    A              ADDRESS OF TVSN-LIST              004
IDBMNT@  DS    A              ADDRESS OF MOUNT-LIST             004
IDBVSQ@  DS    A              ADDRESS OF VSEQ-LIST              004
IDBNUSE1 DS    XL16           -- RESERVED, MUST BE 0 --         004
IDBPRMHP DS    0A             PRIMARY # OF HALFPAGES (SPACE)
IDBFSTHP DS    A              FIRST HALFPAGE (SPACE=ABS)
IDBSCDHP DS    0A             SECONDARY # OF HALFPAGES (SPACE)  004
IDBAMNT  DS    A              NUMBER OF HALFPAGES (SPACE=ABS)   004
IDBPRVL# DS    AL1            NUMBER OF ANY PRIVATE VOLUMES     004
IDBIND7  DS    X              INDICATOR
IDBBLKNL EQU   X'80'           7-7 S BLKSIZE=NULL
IDBKYPNL EQU   X'40'           6-6 S KEYPOS=NULL
```

```
IDBKYLNL EQU   X'20'          5-5 S KEYLEN=NULL
IDBRESNL EQU   X'10'          4-4 S RECSIZE=NULL
IDBFCBNL EQU   X'08'          3-3 S FCBTYPE=NULL
IDBREFNL EQU   X'04'          2-2 S RECFORM=NULL
IDBCODNL EQU   X'02'          1-1 S CODE=NULL
IDBBUFNL EQU   X'01'          0-0 S BUFOFF=NULL                800
IDBIND8  DS    X              INDICATOR
IDBVLPNL EQU   X'80'          7-7 S VALPROP=NULL
IDBLOGNL EQU   X'40'          6-6 S LOGLEN=NULL
IDBVALNL EQU   X'20'          5-5 S VALLEN=NULL
IDBWROUT EQU   X'10'          4-4 S WROUT=YES
IDBIND8U EQU   X'0F'              R -- RESERVED, MUST BE 0 --  950
IDBIND9  DS    X              INDICATOR
IDBDDEVN EQU   X'80'          7-7 S DDEVICE=NOT SPECIFIED
IDBDVOLN EQU   X'40'          6-6 S DVOLUME=NOT SPECIFIED
IDBDSPAN EQU   X'20'          5-5 S DSPACE=NOT SPECIFIED
IDBVLPNS EQU   X'10'          4-4 S VALPROP=NOT SPECIFIED
IDBLOGNS EQU   X'08'          3-3 S LOGLEN=NOT SPECIFIED
IDBVALNS EQU   X'04'          2-2 S VALLEN=NOT SPECIFIED
IDBCODEN EQU   X'02'          1-1 S CODE=NOT SPECIFIED
IDBWRONS EQU   X'01'          0-0 S WROUT=NOT SPECIFIED
IDBIIND  DS    X              ISAM INDICATOR
IDBPREFS EQU   X'80'          7-7 S POOLLNK SPECIFIED          950
IDB2NDDS EQU   X'40'          6-6 S 2ND DSPACE PARAM SPECIFIED 004
IDBDSPAC EQU   X'10'          4-4 S DSPACE=ABS
IDBDVOLU EQU   X'04'          2-2 S DVOLUME=PRIVATE
IDBVMIN  EQU   X'01'          0-0 R MINIMUM FUNCTION FOR ISAM
*                                   VALUE FLAG
IDBVMAX  EQU   X'01'          0-0 S MAXIMUM FUNCTION FOR ISAM
*                                   VALUE FLAG
IDBIINDU EQU   X'2A'              R -- RESERVED, MUST BE 0 --  004
IDBLOGLN DS    AL1            ISAM LOGICAL FLAG LENGTH
IDBVALLN DS    AL1            ISAM VALUE FLAG LENGTH
IDBDEVDL DS    CL8            DEVICE TYPE FOR ISAM DATA VSN    950
IDBCODE  DS    X              CODE FOR TAPE
IDBISOD  EQU   X'20'          5-5 S ISO7D CODE                 200
IDBISO7  EQU   X'10'          4-4 S ISO7 CODE
IDBEBCD  EQU   X'08'          3-3 S EBCDIC CODE
IDBOWN   EQU   X'04'          2-2 S OWN CODE
IDBNDEF  EQU   X'3C'          5-2 R CODE NOT DEFINED           200
IDBCODEU EQU   X'C3'              R -- RESERVED, MUST BE 0 --  200
IDBIGNP  DS    X              IGNORE PROTECTION (TPR)          008
IDBIACC  EQU   X'80'           7-7 S IGNORE ACCESS-RIGHTS      008
IDBIEXD  EQU   X'40'           6-6 S IGNORE EXDATE             008
IDBIRDPW EQU   X'20'           5-5 S IGNORE RDPASS             008
IDBIWRPW EQU   X'10'           4-4 S IGNORE WRPASS             008
IDBIEXPW EQU   X'08'           3-3 S IGNORE EXPASS             008
IDBIGNPU EQU   X'07'               R -- RESERVED MUST BE 0 --  008
```

```
IDBPERF  DS    X                IOPERF-INDICATOR                    050
IDBPFUM  EQU   X'FF'             7-0 S IOPERF=USER-MAX              050
IDBPFNL  EQU   X'FE'             7-1 S IOPERF=NULL                  050
IDBPFVH  EQU   X'03'             1-0 S IOPERF=VERY-HIGH             050
IDBPFHI  EQU   X'02'             1-1 S IOPERF=HIGH                  050
IDBPFST  EQU   X'01'             0-0 S IOPERF=STD                   050
IDBPFNS  EQU   X'00'                 S IOPERF NOT SPECIFIED         050
IDBUSAG  DS    X                IOUSAGE-INDICATOR                   050
IDBUSNL  EQU   X'FF'             7-0 S IOUSAGE=NULL                 050
IDBUSRW  EQU   X'03'             1-0 S IOUSAGE=RDWRT                050
IDBUSWR  EQU   X'02'             1-1 S IOUSAGE=WRITE                050
IDBUSRD  EQU   X'01'             0-0 S IOUSAGE=READ                 050
IDBUSNS  EQU   X'00'                 S IOUSAGE NOT SPECIFIED        050
IDBBUFOF DS    AL1              BUFFER OFFSET VALUE
IDBFSEQ  DS    Y                FSEQ VALUE
IDBTSET  DS    CL4              TSETNAME
IDBFSID  DS    CL6              FIRST VSN FOR TSET ENTRY
IDBVSQVL DS    AL1              VSEQ VALUE                          004
IDBIND10 DS    X
IDBBUFFN EQU   X'80'             7-7 S BUFOFF NOT SPECIFIED
IDBTYPLN EQU   X'40'             6-6 S TYPLAB NOT SPECIFIED
IDBTRANN EQU   X'20'             5-5 S TRANS NOT SPECIFIED
IDBFSEQN EQU   X'10'             4-4 S FSEQ NOT SPECIFIED
IDBSECLN EQU   X'08'             3-3 S SECLEV NOT SPECIFIED
IDBWRCHN EQU   X'04'             2-2 S WRCHK NOT SPECIFIED          751
IDBIN10U EQU   X'03'                 R -- RESERVED, MUST BE 0 --    950
IDBIND11 DS    X
IDBWRCHK EQU   X'80'             7-7 S WRCHK =YES                   751
IDBTRANS EQU   X'40'             6-6 S TRANS =YES
IDBOPR   EQU   X'20'             5-5 S SECLEV = (...,OPR)
IDBBLP   EQU   X'10'             4-4 S SECLEV = LOW                 701
IDBDIN3  EQU   X'08'             3-3 S LABEL=(STD,3)
IDBDIN2  EQU   X'04'             2-2 S LABEL=(STD,2)
IDBDIN1  EQU   X'02'             1-1 S LABEL=(STD,1)
IDBBS2   EQU   X'01'             0-0 S LABEL=(STD,0)
IDBIND12 DS    X
IDBBLPID EQU   X'80'             7-7 S BYPASS LABEL PROCESSING      803
IDBRSDUM EQU   X'40'             6-6 S DUMMY FOR RESTART            750
IDBCKPFE EQU   X'20'             5-5 S CHKPT WHEN FEOV EXECUTED     750
IDBCKPBL EQU   X'10'             4-4 S CHKPT WHEN BLIM OCCURS       750
IDBFSEQV EQU   X'08'             2-2 S FSEQ VALUE SPECIFIED         200
IDBFSNEW EQU   X'04'             1-1 S FSEQ = NEW                   200
IDBFSUNK EQU   X'02'             0-0 S FSEQ = UNK                   200
IDBFSNUL EQU   X'01'             7-7 S FSEQ = NULL                  200
IDBIND13 DS    X
IDBVSEQL EQU   X'80'             7-7 S VSEQ IS A LIST               200
IDBBUFL  EQU   X'40'             6-6 S BUFOFF = L
IDBPOLTP EQU   X'20'             5-5 S POOL TAPE:NO VOLSWITCH       801
```

```
IDBSTRMY EQU   X'10'            4-4 S STREAM=YES                  802
IDBUCON  EQU   X'08'            3-3 S ALLOC: USER CTRL ON  (TPR) 750
IDBUCOFF EQU   X'04'            2-2 S ALLOC: USER CTRL OFF (TPR) 750
IDBL4CHK EQU   X'02'            1-1 S ALLOC: LEVEL 4 CHECK (TPR) 750
IDBL5CHK EQU   X'01'            0-0 S ALLOC: LEVEL 5 CHECK (TPR) 750
IDBIND14 DS    X
IDBTSETN EQU   X'80'            7-7 S TSET NOT SPECIFIED
IDBVSEQN EQU   X'40'            6-6 S VSEQ NOT SPECIFIED
IDBIN14U EQU   X'38'                R -- RESERVED, MUST BE 0 --  220
IDBVOLRU EQU   X'04'            2-2 S VOLUME=REMOVE-UNUSED        220
IDBLKENX EQU   X'02'            1-1 S LOCKENV=XCS                 200
IDBLKENS EQU   X'01'            0-0 S LOCKENV SPECIFIED           200
IDBIND15 DS    X                                                 950
IDBDESCS EQU   X'80'            7-7 S DESTOC  SPECIFIED           950
IDBDESCY EQU   X'40'            6-6 S DESTOC  = YES               950
IDBCMSGS EQU   X'20'            5-5 S CLOSMSG SPECIFIED           950
IDBCMSGY EQU   X'10'            4-4 S CLOSMSG = YES               950
IDBTAPWS EQU   X'08'            3-3 S TAPEWR  SPECIFIED           950
IDBTAPWY EQU   X'04'            2-2 S TAPEWR  = DEVICE-BUFFER     950
IDBX32GS EQU   X'02'            1-1 S EXC32GB SPECIFIED           400
IDBX32GA EQU   X'01'            0-0 S EXC32GB=ALLOWED             400
IDBBLIM  DS    AL3             BLOCK NUMBER LIMIT                750
IDBCHAIN DS    Y               CHAIN VALUE (STREAMING TAPES)     802
IDBBLPPS DS    Y               BYPASS POSITION                   803
IDBBLPVL DS    Y               BYPASS VALUE                      803
IDBPLREF DS    CL8             POOLLNK VALUE                     950
IDBPHPID DS    0A              PRIMARY # OF HALFPAGES (DSPACE)
IDBFSTID DS    A               FIRST HALFPAGE (DSPACE=ABS)       004
IDBSCDID DS    0A              SECONDARY # OF HALFPAGES (DSPACE)
IDBAMTID DS    A               NUMBER OF HALFPAGES (DSPACE=ABS)  004
IDBPRDV# DS    AL1             NUMBER OF ANY PRIVATE DVOLUMES    004
IDBCLOSE DS    X               CLOSE TYPE                        052
IDBCLSLV EQU   X'00'            7-0 R CLOSE=LEAVE                052
IDBCLSDC EQU   X'01'            0-0 S CLOSE=DISCON               052
IDBCLSRP EQU   X'02'            1-1 S CLOSE=REPOS                052
IDBCLSRW EQU   X'03'            1-0 S CLOSE=RWD                  052
IDBCLSIV EQU   X'05'            2,0 S CLOSE=INVAL                052
IDBCLSKD EQU   X'06'            2-1 S CLOSE=KEEP-DATA-IN-CACHE   201
IDBIND16 DS    X                                                 200
IDBWRKFY EQU   X'80'            7-7 S WORKFIL=YES                200
IDBAVHI  EQU   X'40'            6-6 S AVAIL=HIGH                 200
IDBIN16U EQU   X'3E'                 R -- RESERVED, MUST BE 0 --  240
IDBWRKFS EQU   X'01'            0-0 S WORKFIL SPECIFIED          240
IDBIND17 DS    X                                                 200
IDBSTCLS EQU   X'80'            7-7 S STOCLAS SPECIFIED          200
IDBVSETS EQU   X'40'            6-6 S VOLSET  SPECIFIED          200
IDBVSLO  EQU   X'20'            5-5 S ONLY VSETLIST FROM SC (TPR) 200
IDBSCNO  EQU   X'10'            4-4 S STOCLAS=*NONE              202
```

```
IDBVSKSO EQU   X'08'            3-3 S VOLSET: KEEP SO-MIGR  (TPR) 260
IDBIN17U EQU   X'07'                 R -- RESERVED, MUST BE 0 --  260
IDBSTOCL DS    CL8              STOCLAS VALUE                      200
IDBVSET  DS    CL4              VOLSET  VALUE                      200
IDBVSCTL EQU   C'*CVS'          VOLSET=*CONTROL                    230
IDBIND18 DS    X                                                  200
IDBBLSBP EQU   X'80'            7-7 S BLKSIZE BY PROG              200
IDBKYPBP EQU   X'40'            6-6 S KEYPOS  BY PROG              200
IDBKYLBP EQU   X'20'            5-5 S KEYLEN  BY PROG              200
IDBRESBP EQU   X'10'            4-4 S RECSIZE BY PROG              200
IDBFCBBP EQU   X'08'            3-3 S FCBTYPE BY PROG              200
IDBREFBP EQU   X'04'            2-2 S RECFORM BY PROG              200
IDBCODBP EQU   X'02'            1-1 S CODE    BY PROG              200
IDBBUFBP EQU   X'01'            0-0 S BUFOFF  BY PROG              200
IDBIND19 DS    X                                                  200
IDBVLPBP EQU   X'80'            7-7 S VALPROP BY PROG              200
IDBLOGBP EQU   X'40'            6-6 S LOGLEN  BY PROG              200
IDBVALBP EQU   X'20'            5-5 S VALLEN  BY PROG              200
IDBLABBP EQU   X'10'            3-3 S LABEL   BY PROG              200
IDBIN19U EQU   X'0F'                 R -- RESERVED, MUST BE 0 --  200
IDBDSKW  DS    XL1              DISKWR                             202
IDBDWCL  EQU   X'02'            1-1 S DISKWR=BY-CLOSE              202
IDBDWIM  EQU   X'01'            0-0 S DISKWR=IMMEDIATE             202
IDBDWNS  EQU   X'00'                S DISKWR NOT SPECIFIED         202
IDBDATRI DS    X                                                  200
IDBDATRU EQU   X'FE'                 R -- RESERVED, MUST BE 0 --  200
IDBDATFF EQU   X'01'            0-0 S DATATTR FROM FILE            200
IDBDATRF DS    CL54             DATATTR REFERENCE FILENAME         200
IDBIND20 DS    X                                                  210
IDBNEWCE EQU   X'80'            7-7 S CREATE NEW CE (TPR)          210
IDBNDADM EQU   X'40'            6-6 S TSOS PRIVILEGE NEEDED (TPR)  220
IDBNDPHA EQU   X'20'            5-5 S PHYSICAL ALLOC NEEDED (TPR)  220
IDBPFNAJ EQU   X'10'            4-4 S NO IOPERF ADJUST      (TPR)  260
IDBVSUIS EQU   X'08'            3-3 S VOLSETSEL USERID SPEC (TPR)  260
IDBALTFN EQU   X'04'            2-2 S NO ALTERNATIVE FORMAT (TPR)  262
IDBIN20U EQU   X'03'                 R -- RESERVED, MUST BE 0 --  262
IDBVSUID DS    CL8              USERID FOR VOLSET SELECTION (TPR)  260
IDBIND21 DS    X                                                  270
IDBDEFPN EQU   X'80'            7-7 S NO DEFAULT PROTECTION        270
IDBIN21U EQU   X'7F'                 R -- RESERVED, MUST BE 0 --  270
IDBNUSE4 DS    XL4              -- RESERVED, MUST BE 0 --          270
IDBPARLN EQU   *-IDBPFLST        LENGTH OF PARAM LIST               004
```

```
         ERASE MF=D,VERSION=3
         #INTF REFTYPE=REQUEST,INTNAME=ERASE,INTVERS=3
         MFPRE MF=D,DMACID=D,DNAME=ERAPL,PREFIX=IER
IERERAPL DSECT ,
               *,##### PREFIX=IER, MACID=D #####
*
IERERAST DS    0A
*
         FHDR  MF=(C,IER),EQUATES=NO
         DS    0A
IERFHE   DS    0XL8            0   GENERAL PARAMETER AREA HEADER
*
IERIFID  DS    0A              0   INTERFACE IDENTIFIER
IERFCTU  DS    AL2             0   FUNCTION UNIT NUMBER
*                                  BIT 15    HEADER FLAG BIT,
*                                  MUST BE RESET UNTIL FURTHER NOTICE
*                                  BIT 14-12 UNUSED, MUST BE RESET
*                                  BIT 11-0  REAL FUNCTION UNIT NUMBER
IERFCT   DS    AL1             2   FUNCTION NUMBER
IERFCTV  DS    AL1             3   FUNCTION INTERFACE VERSION NUMBER
*
IERRET   DS    0A              4   GENERAL RETURN CODE
IERSRET  DS    0AL2            4   SUB RETURN CODE
IERSR2   DS    AL1             4   SUB RETURN CODE 2
IERSR1   DS    AL1             5   SUB RETURN CODE 1
IERMRET  DS    0AL2            6   MAIN RETURN CODE
IERMR2   DS    AL1             6   MAIN RETURN CODE 2
IERMR1   DS    AL1             7   MAIN RETURN CODE 1
IERFHL   EQU   8               8   GENERAL OPERAND LIST HEADER LENGTH
*
IERPNAM  DC    CL80' '         PATHNAME
*
*
IERACTIO DC    AL1(0)          ACTION -
IERSPCAT EQU   0                  SPACE-CATALOG
IERSPACE EQU   1                  SPACE
IERDATA  EQU   2                  DATA
IERCATLG EQU   3                  CATALOG
IEREXPRT EQU   3                  EXPORT
IERSPCEX EQU   4                  DELETE-OR-EXPORT
*
IERATTR  DC    B'00000000'     ATTRIBUTES -
IERDSTRO EQU   X'80'   7-7 S      DESTROY
IERMTALL EQU   X'40'   6-6 S      MOUNT ALL DISKS
IERMT1ST EQU   X'40'   6-6 R      MOUNT FIRST DISK
IERATTNU EQU   X'3C'   5-2 R   -- UNUSED, MUST BE 0 --              210
IERTBLTI EQU   X'02'   1-1 S      TIME BASE LTI                     210
IERKPACL EQU   X'01'   0-0 S      KEEP ACL
```

```
*
IERIGNOR DC    B'00000000'     IGNORE -
IERIGACC EQU   X'80'   7-7 S      ACCESS
IERIGRTP EQU   X'40'   6-6 S      EXPIRATION DATE
IERIGWRP EQU   X'20'   5-5 S      WRITE PASSWORD
IERIGRDP EQU   X'10'   4-4 S      READ PASSWORD
IERIGEXP EQU   X'08'   3-3 S      EXEC PASSWORD
IERIGNNU EQU   X'07'   2-0 R   -- UNUSED, MUST BE 0 --
*
IEROUTPT DC    B'00000000'     OUTPUT -
IERLIST  EQU   X'80'   7-7 S      LIST ADDRESS IS GIVEN
IERLST@A EQU   X'40'   6-6 R      ABSOLUTE ADDRESS
IERLST@R EQU   X'40'   6-6 S      ADDRESS IN REGISTER
IERLST#A EQU   X'20'   5-5 R      ABSOLUTE LENGTH
IERLST#R EQU   X'20'   5-5 S      LENGTH IN REGISTER
IERLSTSY EQU   X'10'   4-4 S      ERRORS TO SYSOUT
IEROUTNU EQU   X'0F'   3-0 R   -- UNUSED, MUST BE 0 --
*
IERCHECK DC    AL1(0)          CHECK -
IERCKNO  EQU   0                  NO
IERCKERR EQU   1                  ERROR
IERCKSIN EQU   2                  SINGLE
IERCKMUL EQU   3                  MULTIPLE
IERCKUID EQU   4                  USERID
IERCKPVS EQU   5                  PVS
IERCKSTD EQU   255                STD
*
IERFUT1  DC    3X'00'          FOR FUTURE USE, MUST BE ZERO
*
IERLIST@ DC    A(0)            LIST ADDRESS
IERLIST# DC    A(0)            LIST SIZE
*
IERPWARR DC    A(0,0,0)        3 PW'S
*
IERNSTEP DC    Y(0,0,0)        3 ERR'S
*
IERPOSF  DC    B'00000000'     POS-FLAG -
IERPOSB  EQU   X'80'   7-7 S      POS=BEFORE
IERPOSA  EQU   X'40'   6-6 S      POS=AFTER
IERPOSNU EQU   X'3F'   5-0 R   -- UNUSED, MUST BE 0 --
*
IERFUT2  DC    5X'00'          FOR FUTURE USE, MUST BE ZERO
*
         DMSSEL MF=D,PREFIX=IER,VERSION=3
         MFCHK MF=C,SUPPORT=(D,C),PREFIX=I,MACID=XX,DMACID=YYY
         DS    0F
               *,##### PREFIX=I, MACID=XXY #####
*
```

```
*
         DS    0A
*
IERSELI0 DC    B'00000000'     SELECT IND0
IERSIACC EQU   X'80'              ACCESS
IERSIBKP EQU   X'40'              BACKUP-LEVEL
IERSIFCB EQU   X'20'              FCBTYPE
IERSIPWP EQU   X'10'              PASSWORD-PROTECTED
IERSISHR EQU   X'08'              SHARE
IERSICRD EQU   X'04'              CREATION-DATE
IERSIEXD EQU   X'02'              EXPIRATION-DATE
IERSILAD EQU   X'01'              LAST-ACCESS-DATE
*
IERSELI1 DC    B'00000000'     SELECT IND1
IERSIEXT EQU   X'80'              EXTENTS
IERSIFSZ EQU   X'40'              FREESIZE
IERSISIZ EQU   X'20'              SIZE
IERSISUP EQU   X'10'              SUPPORT
IERSIVOL EQU   X'08'              VOLUME
IERSIACL EQU   X'04'              ACL
IERSIACT EQU   X'02'              ACCESS-COUNT
IERSIBAC EQU   X'01'              BASIC-ACL
*
                                                  15:29:06  2001-10-15   PAGE
0041
SOURCE STATEMENT
IERSELI2 DC    B'00000000'     SELECT IND2
IERSIBKC EQU   X'80'              BLOCK-CONTROL
IERSIBCT EQU   X'40'              BLOCK-COUNT
IERSIDWR EQU   X'20'              DISK-WRITE
IERSIGUA EQU   X'10'              GUARDS
IERSIIOP EQU   X'08'              IO-PERFORMANCE
IERSILCD EQU   X'04'              LAST-CHANGE-DATE
IERSIIOU EQU   X'02'              IO-USAGE
IERSILPA EQU   X'01'              LAST-PAGE
*
IERSELI3 DC    B'00000000'     SELECT IND3
IERSIMIG EQU   X'80'              MIGRATE
IERSIPRO EQU   X'40'              PROTECTION-LEVEL
IERSIRSP EQU   X'20'              RELEASE-SPACE
IERSISTA EQU   X'10'              STATE
IERSISTL EQU   X'08'              STORAGE-LEVEL
IERSITYP EQU   X'04'              TYPE
IERSICCS EQU   X'02'              CCS
IERSI3UU EQU   X'01'             -- UNUSED, MUST BE 0 --
*
IERSELI4 DC    B'00000000'     SELECT IND4                              200
IERSIVST EQU   X'80'              VOLUME-SET                           200
```

```
IERSIWKF EQU   X'40'              WORK-FILE                          200
IERSIAVA EQU   X'20'              AVAILABILITY                       200
IERSISOM EQU   X'10'              SO-MIGRATION                       200
IERSIDLD EQU   X'08'              DELETION-DATE                      200
IERSIUIN EQU   X'04'              USER-INFO                          200
IERSIAIN EQU   X'02'              ADM-INFO                           200
IERSI4UU EQU   X'01'            -- UNUSED, MUST BE 0 --              200
*
IERSELI5 DC    B'00000000'      SELECT IND5                          200
IERSISCL EQU   X'80'              STORAGE-CLASS                      200
IERSIMCL EQU   X'40'              MANAGEMENT-CLASS                   200
IERSIPFO EQU   X'20'              FILE-PREFORMAT                     220
IERSI5UU EQU   X'1F'            -- UNUSED, MUST BE 0 --              220
*
IERUNUS1 DC    X'0000'                                               200
*
IERACC   DC    A(0)             ACCESS-COUNT  - FROM
IERACC2  DC    A(0)                           - TO
*
IERBACL  DC    B'00000000'      BASIC-ACL
IERBACLY EQU   X'80'               YES
IERBACLN EQU   X'40'               NONE
IERBACLU EQU   X'2F'            -- UNUSED, MUST BE 0 --
*
IERBOW   DC    B'00000000'      BASIC-ACL-OWNER
IERBOWRS EQU   X'80'               READ-RIGHT-SPECIFIED
IERBOWWS EQU   X'40'               WRITE-RIGHT-SPECIFIED
IERBOWXS EQU   X'20'               EXEC-RIGHT-SPECIFIED
IERBOWRY EQU   X'10'               READ = YES
IERBOWWY EQU   X'08'               WRITE = YES
IERBOWXY EQU   X'04'               EXEC = YES
IERBOWNO EQU   X'02'               NO-ACCESS
IERBOWUU EQU   X'01'             -- UNUSED, MUST BE 0 --
*
IERBGR   DC    B'00000000'      BASIC-ACL-GROUP
IERBGRRS EQU   X'80'               READ-RIGHT-SPECIFIED
IERBGRWS EQU   X'40'               WRITE-RIGHT-SPECIFIED
IERBGRXS EQU   X'20'               EXEC-RIGHT-SPECIFIED
IERBGRRY EQU   X'10'               READ = YES
IERBGRWY EQU   X'08'               WRITE = YES
IERBGRXY EQU   X'04'               EXEC = YES
IERBGRNO EQU   X'02'               NO-ACCESS
IERBGRUU EQU   X'01'             -- UNUSED, MUST BE 0 --
*
IERBOT   DC    B'00000000'      BASIC-ACL-OTHERS
IERBOTRS EQU   X'80'               READ-RIGHT-SPECIFIED
IERBOTWS EQU   X'40'               WRITE-RIGHT-SPECIFIED
IERBOTXS EQU   X'20'               EXEC-RIGHT-SPECIFIED
```

```
IERBOTRY EQU   X'10'              READ = YES
IERBOTWY EQU   X'08'              WRITE = YES
IERBOTXY EQU   X'04'              EXEC = YES
IERBOTNO EQU   X'02'              NO-ACCESS
IERBOTUU EQU   X'01'            -- UNUSED, MUST BE 0 --
*
IERBKL   DC    B'00000000'      BACKUP LEVEL FLAG
IERBKLA  EQU   X'80'   7-7 S      A
IERBKLB  EQU   X'40'   6-6 S      B
IERBKLC  EQU   X'20'   5-5 S      C
IERBKLD  EQU   X'10'   4-4 S      D
IERBKLE  EQU   X'08'   3-3 S      E
IERBKLNU EQU   X'07'   2-0 R    -- UNUSED, MUST BE 0 --
*
IERBLKCT DC    B'00000000'      BLKCTRL -
IERBCTNN EQU   X'80'   7-7 S      NONE
IERBCTPK EQU   X'40'   6-6 S      PAMKEY
IERBCTDA EQU   X'20'   5-5 S      DATA2K
IERBCTNO EQU   X'10'   4-4 S      NO
IERBCTD4 EQU   X'08'   3-3 S      DATA4K
IERBCTN2 EQU   X'04'   2-2 S      NK2
IERBCTN4 EQU   X'02'   1-1 S      NK4
IERBCTNU EQU   X'01'   0-0 R    -- UNUSED, MUST BE 0 --
*
IERFUT3  DC    X'0000'          SPACE FOR FUTURE USE, MUST BE 0
*
IERBLCT  DC    A(0)             BLOCK-COUNT  - FROM
IERBLCT2 DC    A(0)                          - TO
*
IERCRD   DC    CL10' '          CREATION DATE    - FROM DATE
IERCRT   DC    CL8' '                                 TIME
IERCRD2  DC    CL10' '                          - TO DATE
IERCRT2  DC    CL8' '                               TIME
*
IEREXD   DC    CL10' '          EXPIRATION DATE  - FROM DATE
IEREXT   DC    CL8' '                                 TIME
IEREXD2  DC    CL10' '                          - TO DATE
IEREXT2  DC    CL8' '                               TIME
*
IEREXS   DC    A(0)             EXTENTS  - FROM
IEREXS2  DC    A(0)                      - TO
*
IERFSIZE DC    A(0)             FSIZE  - FROM
IERFSIZ2 DC    A(0)                    - TO
*
IERDW    DC    B'00000000'      DISK WRITE -
IERDWIM  EQU   X'80'   7-7 S      IMMEDIATE
IERDWBC  EQU   X'40'   6-6 S      BY-CLOSE
```

```
IERDWNU  EQU   X'3F'   5-0 R   -- UNUSED, MUST BE 0 --
*
IERFCB   DC    B'00000000'     FCB TYPE FLAG -
IERFCBP  EQU   X'80'   7-7 S      PAM
IERFCBS  EQU   X'40'   6-6 S      SAM
IERFCBI  EQU   X'20'   5-5 S      ISAM
IERFCBB  EQU   X'10'   4-4 S      BTAM
IERFCBN  EQU   X'08'   3-3 S      NONE
IERFCBNU EQU   X'07'   2-0 R   -- UNUSED, MUST BE 0 --
*
IERGUA   DC    B'00000000'     GUARDS SPECIFIED FLAG
IERGUARS EQU   X'80'   7-7 S      READ-SPECIFIED
IERGUAWS EQU   X'40'   6-6 S      WRITE-SPECIFIED
IERGUAXS EQU   X'20'   5-5 S      EXEC-SPECIFIED
IERGUUNU EQU   X'10'   4-4 S      UNUSED, MUST BE ZERO --           200
IERGUANS EQU   X'08'   3-3 S      GUARDS=NONE SPECIFIED
IERGUAYS EQU   X'04'   2-2 S      GUARDS=YES SPECIFIED
IERGUAUU EQU   X'03'   1-0 R   -- UNUSED, MUST BE ZERO --
IERGUAR  DC    CL18' '            GUARDS-READ
IERGUAW  DC    CL18' '            GUARDS-WRITE
IERGUAX  DC    CL18' '            GUARDS-EXEC
*
IERIP    DC    B'00000000'     IO-PERFORMANCE
IERIPVH  EQU   X'80'   7-7 S      VERY-HIGH
IERIPHI  EQU   X'40'   6-6 S      HIGH
IERIPST  EQU   X'20'   5-5 R      STD
IERIPNU  EQU   X'1F'   4-0 R   -- UNUSED, MUST BE 0 --
*
IERIU    DC    B'00000000'     IO-USAGE
IERIURW  EQU   X'80'   7-7 S      READ-WRITE
IERIURD  EQU   X'40'   6-6 S      READ
IERIUWR  EQU   X'20'   5-5 R      WRITE
IERIUNU  EQU   X'1F'   4-0 R   -- UNUSED, MUST BE 0 --
*
IERUNUS2 DC    X'00'
*
IERLAD   DC    CL10' '         LAST ACCESS DATE - FROM DATE
IERLAT   DC    CL8' '                             TIME
IERLAD2  DC    CL10' '                        - TO DATE
IERLAT2  DC    CL8' '                             TIME
*
IERLAPA  DC    A(0)            LAST-PAGE  - FROM
IERLAPA2 DC    A(0)                       - TO
*
IERLCD   DC    CL10' '         LAST CHANGE DATE - FROM DATE
IERLCT   DC    CL8' '                             TIME
IERLCD2  DC    CL10' '                        - TO DATE
IERLCT2  DC    CL8' '                             TIME
```

```
*
IERDLD   DC    CL10' '            DELETION DATE - FROM DATE         200
IERDLT   DC    CL8' '                                 TIME          200
IERDLD2  DC    CL10' '                          - TO DATE           200
IERDLT2  DC    CL8' '                               TIME            200
*
IERMIGRT DC    B'00000000'     MIGRATE -
IERMGALL EQU   X'80'   7-7 S      ALLOWED
IERMGINH EQU   X'20'   5-5 S      INHIBIT
IERMGFOB EQU   X'10'   4-4 S      FORBIDDEN                         200
IERMGANY EQU   X'B0'   7,4 R      ANY                               200
IERMIGNU EQU   X'4F'   3-0 R   -- UNUSED, MUST BE 0 --              200
*
IERPWP   DC    B'00000000'     PW PROTECT FLAG -
IERPWPRD EQU   X'80'   7-7 S      READ
IERPWPWR EQU   X'40'   6-6 S      WRITE
IERPWPEX EQU   X'20'   5-5 S      EXEC
IERPWPNO EQU   X'10'   4-4 S      NONE
IERPWPNU EQU   X'0F'   3-0 R   -- UNUSED, MUST BE 0 --
*
IERPROL  DC    B'00000000'     PROTECTION-LEVEL-FLAG
IERPROLO EQU   X'80'   7-7 S      LEVEL 0
IERPROL1 EQU   X'40'   6-6 S      LEVEL 1
IERPROL2 EQU   X'20'   5-5 S      LEVEL 2
IERPROUU EQU   X'1F'   4-0 R   -- UNUSED, MUST BE 0 --
*
IERRLSP  DC    B'00000000'     RELEASE-SPACE-FLAG
IERRLSPA EQU   X'80'   7-7 S      ALLOWED
IERRLSPI EQU   X'40'   6-6 S      IGNORED
IERRLSPU EQU   X'3F'   5-0 R   -- UNUSED, MUST BE 0 --              200
*
IERSELFO DC    B'00000000'     SELECT FLAG 0 -
IERTYPEF EQU   X'80'   7-7 S      TYPE = FILE
IERTYPEG EQU   X'40'   6-6 S      TYPE = FGG
IERTYPEA EQU   X'C0'   7-6 S      TYPE = ANY
IERTYPEP EQU   X'20'   5-5 S      TYPE = PLAM
IERSTCNS EQU   X'10'   4-4 S      STATE/CACHE NOT SAV
IERAVAST EQU   X'08'   3-3 S      AVAILABILITY = STD                200
IERAVAHI EQU   X'04'   2-2 S      AVAILABILITY = HIGH               200
IERSFONU EQU   X'03'   1-0 R   -- UNUSED, MUST BE 0 --              200
*
IERSELF1 DC    B'00000000'     SELECT FLAG 1 -
IERACCRD EQU   X'80'   7-7 S      ACCESS = READ
IERACCWR EQU   X'40'   6-6 S      ACCESS = WRITE
IERSHARY EQU   X'20'   5-5 S      SHARE  = YES
IERSHARN EQU   X'10'   4-4 S      SHARE  = NO
IERSHARS EQU   X'08'   3-3 S      SHARE  = SPECIAL
IERACLY  EQU   X'04'   2-2 S      ACL    = YES
```

```
IERACLN  EQU   X'02'   1-1 S      ACL    = NO
IERSF1NU EQU   X'01'   0-0 R   -- UNUSED, MUST BE 0 --
*
IERSELF2 DC    B'00000000'     SELECT FLAG 2 -
IERSEQFS EQU   X'80'   7-7 S      SIZE   = FSIZE
IERFSEQS EQU   X'40'   6-6 S      FSIZE  = SIZE
IERCRDNO EQU   X'20'   5-5 S      CRDATE = NONE
IEREXDNO EQU   X'10'   4-4 S      EXDATE = NONE
IERLADNO EQU   X'08'   3-3 S      LADATE = NONE
IERLCDNO EQU   X'04'   2-2 S      LCDATE = NONE
IERDLDNO EQU   X'02'   1-1 S      DELDATE = NONE                    200
IERSF2UU EQU   X'01'   0-0 R   -- UNUSED, MUST BE 0 --              200
*
IERSTA   DC    B'00000000'     STATE FLAG -
IERSTNOC EQU   X'80'   7-7 S      NOCLOS
IERSTCLO EQU   X'40'   6-6 S      CLOSED
IERSTREP EQU   X'20'   5-5 S      REPAIR NEEDED
IERSTCAC EQU   X'10'   4-4 S      CACHED
IERSTNCA EQU   X'08'   3-3 S      NOT CACHED
IERSTNOA EQU   X'04'   2-2 S      NO OPEN ALLOWED
IERSTOPA EQU   X'02'   1-1 S      OPEN ALLOWED
IERSTDFR EQU   X'01'   0-0 S      DEFECT-REPORTED
*
IERSTLEV DC    B'00000000'     SLEVEL -
IERSLVL0 EQU   X'80'   7-7 S      S0
IERSLVL1 EQU   X'40'   6-6 S      S1
IERSLVL2 EQU   X'20'   5-5 S      S2
IERSLANY EQU   X'E0'   7-5 R      ANY
IERSLVNU EQU   X'1F'   4-0 R   -- UNUSED, MUST BE 0 --
*
IERSUP   DC    B'00000000'     SUPPORT TYPE FLAG -
IERSUPPU EQU   X'80'   7-7 S      PUBLIC
IERSUPPR EQU   X'40'   6-6 S      PRDISC
IERSUPTP EQU   X'20'   5-5 S      TAPE
IERSUPNU EQU   X'1F'   4-0 R   -- UNUSED, MUST BE 0 --
*
IERSMS   DC    B'00000000'     SMS FLAG -                        200
IERSOMAL EQU   X'80'   7-7 S      SO-MIGRATION = ALLOWED         200
IERSOMFO EQU   X'40'   6-6 S      SO-MIGRATION = FORBIDDEN       200
IERWKFYE EQU   X'20'   5-5 S      WORK-FILE = YES                200
IERVCTRL EQU   X'10'   4-4 S      VOL-SET = CONTROL              210
IERSMSNU EQU   X'0F'   3-0 R   -- UNUSED, MUST BE 0 --           210
*
IERUNUS3 DC    X'00'                                             200
*
IERSIZE  DC    A(0)            SIZE  - FROM
IERSIZE2 DC    A(0)                  - TO
*
```

```
IERVOL   DC    CL6' '          VOLUME
IERCCS   DC    CL8' '          CODED-CHARACTER-SET
IERVOLST DC    CL4' '          VOLUME-SET (SMS)                    200
IERUSINF DC    CL8' '          USER-INFORMATION                    200
IERADINF DC    CL8' '          ADM-INFORMATION                     200
IERMGMCL DC    CL8' '          MANAGEMENT-CLASS                    200
IERSTOCL DC    CL8' '          STORAGE-CLASS                       200
IERPFORM DC    B'00000000'     FILE-PREFORMAT                      220
IERPFONO EQU   X'80'   7-7 S      NONE                             220
IERPFOK  EQU   X'40'   6-6 S      K                                220
IERPFON2 EQU   X'20'   5-5 S      NK2                              220
IERPFON4 EQU   X'10'   4-4 S      NK4                              220
IERPFOAN EQU   X'F0'   7-4 R      ANY                              220
IERPFOUN EQU   X'0F'   3-0 R   -- UNUSED, MUST BE 0 --             220
IERFUT4  DC    61X'00'         SPACE FOR FUTURE USE, MUST BE 0
IERPLLN  EQU   *-IERERAST      LENGTH
*
```

```
         IDFCB D
*
******** FCB (P1 REGION)  **********************************
*
         #INTF REFTYPE=REQUEST,                                        C
               INTNAME=IDFCB,INTCOMP=1
         MFPRE MF=D,PREFIX=,MACID=D1,DNAME=D1FCB,ALIGN=F            200
ID1FCB   DSECT ,
               *,##### PREFIX=I, MACID=D1 #####
*
ID1P1FST EQU   *
ID1CONAR DS    0F
ID1IOAR1 DS    F                 ADDR OF IOAREA1               ISQP
ID1IOA1N EQU   X'80'             S IOAREA1=NO                  ISQP
ID1IOA1B EQU   X'40'             S IOAREA1=                    ISQP
ID1IOAR2 DS    F                 ADDR OF IOAREA2               ISQP
ID1IOA2N EQU   X'80'             S IOAREA2=NO                  ISQP
ID1IOA2B EQU   X'40'             S IOAREA2=                    ISQP
ID1EXIT  DS    F                 ADDR OF EXIT LIST             ISQP
ID1NOXIT EQU   X'80'             S EXIT=                       ISQP
ID1MSTXT EQU   X'40'             S MASTER EXIT LIST            ISQP
ID1CLBT  EQU   X'20'             S EXIT MODIFY BIT             ISQP
ID1KEYAR DS    F                 ADDR OF KEY                    S
ID1KEYAN EQU   X'80'             S KEYARG=                      S
*
ID1PASS  DS    XL4               PASSWORD                      ISQPM
ID1RETPD DS    H                 RETENTION PERIOD              ISQPM
ID1RECSI DS    H                 RECORD SIZE                   IS  M
ID1BLKSI DS    H                 BLOCK SIZE                    IS PM
ID1BLSTD EQU   X'80'             S BLKSIZE=STD                 IS  M
ID1KEYPO DS    H                 KEY POSITION                   S  M
ID1KEYLE DS    HL1               KEY LENGTH                     S  M
ID1INDEX DS    HL1               INDEX                          S  M
ID1PAD   DS    HL1               I.S. PADDING                   S  M
*
ID1OPEN  DS    X                 OPEN TYPE :                   ISQPM
ID1SINOT EQU   X'80'             S SINOUT                         P
ID1OUTIN EQU   X'40'             S OUTIN                       IS PM
ID1INOUT EQU   X'20'             S INOUT                       IS PM
ID1UPDAT EQU   X'10'             S UPDATE                      I Q M
ID1EXTEN EQU   X'08'             S EXTEND                      IS PM
ID1OUTPU EQU   X'04'             S OUTPUT                      ISQPM
ID1REVER EQU   X'02'             S REVERSE                     I   M
ID1INPUT EQU   X'01'             S INPUT                       ISQPM
*
ID1INDO  DS    X                 INDICATOR 0 :
ID1IIND  EQU   ID1INDO           ISAM INDICATOR
ID1VLPNL EQU   X'08'             S VALPROP=NULL
```

```
ID1LOGNL EQU   X'04'              S LOGLEN=NULL
ID1VALNL EQU   X'02'              S VALLEN=NULL
ID1WROUT EQU   X'01'              S WROUT=YES
                                                15:29:06  2001-10-15   PAGE
0047
SOURCE STATEMENT
ID1FRRC  EQU   X'80'              S RECON=YES
ID1FRRS  EQU   X'40'              S RESET=YES
ID1IPP   EQU   X'20'              S VALPROP/VALLEN/LOGLEN SPEC.
*
ID1IND1  DS    X                  INDICATOR 1 :                  ISQPM
ID1TRANS EQU   X'80'              S TRANSLATE TAPE
ID1LABEL EQU   X'60'              FILE LABEL OPTIONS             ISQPM
ID1STD   EQU   X'60'              R STD LABELS                   ISQPM
ID1NSTD  EQU   X'40'              S NSTD LABELS                  I  PM
ID1LABL  EQU   X'20'              S LABEL=NO
ID1AMN   EQU   X'18'              PRINTER CONTROL                ISQPM
ID1A     EQU   X'18'              S ASA CONTROL CHAR             ISQPM
ID1M     EQU   X'08'              S EBCDIC CONTROL               ISQPM
ID1N     EQU   X'18'              R NO CONTROL CHAR              ISQPM
ID1RECFO EQU   X'06'              RECORD FORM                    ISQPM
ID1F     EQU   X'04'              S FIXED LENGTH                 ISQPM
ID1V     EQU   X'02'              S VARIABLE LENGTH              IS  M
ID1U     EQU   X'06'              S UNDEFINED                    IS PM
ID1TPMRK EQU   X'01'              S TPMARK=YES
*
ID1IND2  DS    X                  INDICATOR 2 : (SET,RESET)
ID1FCBTY EQU   X'C0'              FCB TYPE                       ISQPM
ID1SAM   EQU   X'C0'              R SAM                          I   M
ID1ISAM  EQU   X'40'              S ISAM                          S  M
ID1PAM   EQU   X'C0'              S PAM                            Q M
ID1BTAM  EQU   X'80'              S BTAM                            PM
ID1DUPKE EQU   X'20'              S DUP KEY ALLOWED               S  M
ID1OVERL EQU   X'10'              S OVERLAP=YES                   S  M
ID1INDPR EQU   X'08'              S INDEX=(ABSEXP)                S  M
ID1SHB   EQU   X'04'              S SPECIAL HANDLING BIT           Q
ID1FORM  EQU   X'02'              S FORM=SHORT                     QP
ID1VMIN  EQU   X'01'              R MINIMUM FUNCT.FOR ISAM VALUE FLAG
ID1VMAX  EQU   X'01'              S MAXIMUM FUNCT.FOR ISAM VALUE FLAG
*
ID1IND3  DS    X                  INDICATOR 3 :
ID1NULLP EQU   X'FF'              NULL FILE CHARACTERISTIC PARAMETERS
ID1RECSN EQU   X'80'              S RECSIZE=,
ID1BLKSN EQU   X'40'              S BLKSIZE=,
ID1KEYPN EQU   X'20'              S KEYPOS=,
ID1KEYLN EQU   X'10'              S KEYLEN=,
ID1RECFN EQU   X'08'              S RECFORM=,
ID1FCBTN EQU   X'04'              S FCBTYPE=,
```

```
ID1CODEN EQU   X'02'             S CODE=,
ID1BUFNL EQU   X'01'             S BUFOFF3=,
*
ID1IOREG DS    HL1               IOREG                          IS
ID1VARBL DS    HL1               VARBLD                         IS
ID1LINK  DS    CL8               FCB/LINK NAME                  ISQPM
ID1FILE  DS    CL54              FILE NAME                      ISQPM
ID1VARAR DS    0F
ID1P2LNK DS    F                 LINK TO P2 REGION              ISQP
ID1UFCBS EQU   *-ID1CONAR        SIZE OF CONSTANT REGION
*
ID1COMMN DS    11F               COMMON REGION                  ISQP
ID1CMEND EQU   *                 END COMMON REGION ------------------
*
         ORG   ID1COMMN          ***  S A M  ***                I
ID1LHECN DS    F                 LHECON CURRENT                 I
ID1AFNAL DS    F                 LHECON MAXIMUM                 I
ID1BLKCT DS    PL4               BLOCK COUNT                    I
ID1RPTR  DS    F                 LOG BUFF/RCD PTR               I
ID1RGSAV DS    5F                SAVE AREA                      I
ID1TRADR DS    A                 A(USER TRANS TAB)  READ MODE  I  P
ID1TRADW DS    A                 A(USER TRANS TAB) WRITE MODE  I  P
*
         ORG   ID1COMMN          ACCESS METHOD ISAM
ID1INHK@ DS    A                 @ HIGHKEY
ID1INNR@ DS    A                 @ NEXT RECORD
ID1INCR@ DS    A                 @ CURRENT RECORD
ID1INFLO DS    X                 FLAGB. D1SUSNT: WIE ISAM-ALT
ID1INSTF EQU   X'01'             D1SHAR1 ! WIE ISAM-ALT !
ID1INLTF EQU   X'04'             D1DLOK1 ! WIE ISAM-ALT !
ID1INLSP EQU   X'10'             D1DLOK  ! WIE ISAM-ALT !
ID1INFL1 DS    X
ID1INHK  EQU   X'01'             @ HIGHKEY VALID
ID1INFL2 DS    X                 UNBENUTZT
ID1INSTA DS    X                 P1LOGICAL-STATUS
ID1INNS  EQU   X'01'             NOT_SEQUENTIAL
ID1INSR  EQU   X'02'             SEQUENTIAL_READ (GET/GETR MIT IOAREA)
ID1INGT  EQU   X'04'             SEQUENTIAL_READ_GET
ID1INGR  EQU   X'08'             SEQUENTIAL_READ_GETR
ID1INSL  EQU   X'10'             SEQUENTIAL_LOAD (PUT MIT IOAREA)
ID1INPC  EQU   X'20'             PUTX_LOCATE_TO_BE_CLOSED
ID1INSPB EQU   X'40'             SEQ_READ_AFTER_PTXLCB
ID1INPCB EQU   X'80'             PUTX_LOC_TO_BE_CLOSED_BLK
ID1INMRL DS    CL4               MINIMALE RECORD-LAENGE
ID1INPOL DS    H                 SUMME UEBER POINTERLAENGEN BEI PUT
ID1INVLN DS    X                 ENTSPRICHT  D1VALLN ! WIE ISAM-ALT !
ID1INLLN DS    X                 ENTSPRICHT  D1LOGLN ! WIE ISAM-ALT !
ID1INTBS DS    A                 BLKSIZE OHNE TRAILER (N*2032 - 12)
```

```
ID1INPBS DS    A                 BLKSIZE OHNE PAD-SIZE
ID1INBS  DS    A                 BLKSIZE = BLOCKFAKTOR * 2048
ID1PRECS DS    H               RECSIZE DES VORGAENGER PUT-LOCATE   951
*
         ORG   ID1COMMN          ***  I S A M  ***              S
ID1GRPTR DS    A(0)              'GETR' ADDRESS                S
ID1GPTR  DS    A(0)              'GET' ADDRESS                 S
ID1GETSW DS    0F                GET | GETR SWITCHES           S
ID1SETL  EQU   X'80'             S INDICATE SETL               S
ID1GET   EQU   X'40'             S INHIBIT GET EOB             S
ID1GETR  EQU   X'20'             S INHIBIT GETR EOB            S
ID1GETER EQU   X'10'             S ERROR DURING GET            S
ID1NULLF EQU   X'08'             S NULL FILE                   S
ID1FIRST DS    A(0)              'FIRST' RECORD POINTER        S
*
ID1SUSNT DS    X                 SHARED UPDATE SENTINEL :
ID1SHAR1 EQU   X'01'             S SHARUPD=YES FOR THIS FILE
ID1SHAR2 EQU   X'02'             S SHARUPD=YES FOR SOME FILE
ID1DLOK1 EQU   X'04'             S DATA PG LOCKED THIS FILE
ID1DLOK2 EQU   X'08'             S DATA PG LOCKED SOME FILE
ID1DLOK  EQU   X'10'             S LOCK SPECIFIED IN MACRO
*                                R NOLOCK SPECIFIED IN THE MACRO
ID1PGLOK EQU   X'20'             S GET,GETR,GETFL ARE ILLEGAL,
*                                  SET WHEN USER TAKES PGLOCK EXIT
ID1ISLK  EQU   X'40'             S SAM LOCK EXISTS
ID1SWEAK EQU   X'80'             S SHARUPD=WEAK FOR THIS FILE
*
         ORG   ID1SUSNT
ID1CURBF DS    A(0)              CURRENT BUFFER ADDRESS        S
ID1PUTSW DS    0F                PUT SWITCH                    S
ID1PUT   EQU   X'80'             S EXTENDED 'PUT'              S
ID1FPUT  EQU   X'40'             S PUT TO EMPTY BUFFER
ID1NPUT  EQU   X'20'             S PUT TO EMPTY PAGE
ID1SPUT  EQU   X'10'             S SEQ CHK NOT ALLOWED
ID1PPTR  DS    A(0)              'PUT' POINTER                 S
*
ID1SPREM EQU   ID1PPTR
ID1SWORK DS    H                 ALIGNMENT CONVERSION          S
ID1VALLN DS    X                 ISAM VALUE FLAG LENGTH
ID1LOGLN DS    X                 ISAM LOGICAL FLAG LENGTH
ID1REGSV DS    4F                REGISTER SAVE AREA            S
ID1CMPD  DS    1F                LOCATE MODE COMPARAND
*
         ORG   ID1COMMN          ***  P A M  ***
ID1LWB   DS    F                 LAST WAITED BUFFER             Q
ID1NOWT  EQU   X'FF'              NO BUFFER WAITED              Q
         ORG   ID1COMMN          RFA PAM FIELD
ID1PLPK  DS    H                 # OF LOCKED PAM PAGES
```

```
ID1PUPK  DS    H                 # OF UNLOCKED PAM PAGES
ID1KEY1  DS    CL16              KEY                              Q
ID1KEY2  DS    CL16              KEY                              Q
ID1REQ   DS    0H                # PAM REQUESTS                   Q
ID1NBPP  DS    CL1               # PROCESSED HP'S                 Q
ID1PRQ   DS    CL1               PAMREQS                          Q
ID1TOUT  DS    AL2               PAM TIME OUT                     Q
ID1CHERR DS    A(0)              CHAIN ELEMENT IN ERR             Q
*
         ORG   ID1COMMN          ***  B T A M  ***                 P
*              BTAM ALSO USES FIELDS &P.TRADR AND &P.TRADW
*              OF THE SAM WORK AREA
         ORG   ID1CMEND
*
********       COMMON TO ALL ACCESS METHODS       ********************
*
ID1RTNAD DS    0F                RETURN ADDR                    ISQP
ID1LGINF DS    A(0)              ADDRESS OF LOGINFO
ID1ECB   DS    XL2               ERROR CODE                     ISQP
*
ID1XITB  DS    X                 ERROR EXIT BYTE :              ISQP
ID1OPENX EQU   4                 OPEN EXIT                      ISQP
ID1OPENE EQU   8                 OPEN ERROR                     ISQP
ID1PASSE EQU   12                PASSWORD ERROR                 ISQP
ID1LOCK  EQU   16                LOCKED FILE                    ISQP
ID1NODEV EQU   20                NO DEVICE                      ISQP
ID1OPENZ EQU   24                OPEN EXIT AFTER LABEL          ISQP
ID1OPENV EQU   28                NON-STANDARD TAPE LABELS
ID1RECNX EQU   32                RECON EXIT
ID1LABGN EQU   36                GEN/RD HDR LAB VOL1            I  P
ID1LABEO EQU   40                GEN/RD EOV LAB                 I  P
ID1CLOSE EQU   44                CLOSE ERROR
ID1LABEN EQU   48                GEN/RD EOF LAB                 I  P
ID1EOVCT EQU   52                VOLUME SWAPPING DONE           I  P
ID1PLKXT EQU   56                LOCKED PAGE(S) EXIT             SQ
ID1DEDLK EQU   60                DEADLOCK EXIT                   SQ
ID1EOFAD EQU   64                EOF ADDR                       IS P
ID1ERRAD EQU   68                ERROR ADDR                     ISQP
ID1ERROP EQU   72                READ ERROR OPTIONS             I
ID1NOSPA EQU   76                FOR SECONDARY ALLOC
ID1ISPER EQU   80                INSUFFICIENT SPACE              S
ID1DUPEK EQU   84                DUPLICATE KEY                   S
ID1NOFIN EQU   88                RECORD NOT FOUND                S
ID1USERE EQU   92                USER ERROR                     ISQP
ID1SEQCH EQU   96                SEQUENCE ERROR                  S
ID1WLRER EQU   100               WRONG LENGTH RECORD
ID1OPENC EQU   104               OPENC EXIT
ID1LABER EQU   108               LABERR EXIT
```

```
ID1CLOSP EQU   112               CLOSPOS EXIT
*
ID1STAT  DS    0CL5              STATUS BYTES                    QP
ID1SDB   DS    C                 STANDARD DEVICE BYTE            QP
ID1SB1   DS    C                 SENSE BYTE 1                    QP
ID1SB2   DS    C                 SENSE BYTE 2                    QP
ID1SB3   DS    C                 SENSE BYTE 3                    QP
ID1EFB   DS    C                 EXECUTIVE FLAG BYTE             QP
*
ID1SNT1  DS    X                 SENTINEL1 : (SET,RESET)       ISQP
ID1REMOT EQU   X'80'             S FILE ACCESS IN RFA MODE
ID1BLOGS EQU   X'40'             S SAM BUFFER LOGGED
ID1NEWV  EQU   X'20'             S EXTENSION FLAG
ID1CHECK EQU   X'01'             S NORMAL CHECK COMPLETION
ID1SYCLS EQU   X'02'             S SYSTEM FILE
*
ID1CODE  EQU   ID1SNT1           CODE OF TAPE :
ID1ISO7  EQU   X'10'             S ISO7 CODE
ID1EBCD  EQU   X'08'             S EBCDI CODE
ID1OWN   EQU   X'04'             S OWN CODE
ID1NDEF  EQU   X'1C'             R CODE NOT DEFINED
*
ID1SNT2  DS    X                 SENTINEL2 : (SET,RESET)
ID1UPACT EQU   X'08'             S UPDATED BLOCK               I
ID1RAW   EQU   X'04'             S READ AFTER WRITE
ID1OPN   EQU   X'40'             S FILE OPEN                   ISQP
ID1GSET  EQU   X'20'             S LAST ACTION (GET)            S
*                                S SETL=V|B                     S
*                                R SETL=E                       S
*                                S ELIM                         S
ID1GRSET EQU   X'18'             S LAST ACTION (GETR)           S
ID1GRST1 EQU   X'10'             S SETL=V(V FOUND)              S
*                                R SETL=V(V N/FOUND)            S
*                                R SETL=B                       S
*                                R ELIM                         S
ID1LPUTX EQU   X'08'             S LOCATE PUTX                  S
ID1EXCAT EQU   X'02'             S EXCAT IN PROCESS
*
ID1DFCBS EQU   *-ID1VARAR        SIZE OF VARIABLE REGION
ID1VIND  DS    H                 EXTENTION LENGTH
         DS    0F
ID1FCBS  EQU   *-ID1CONAR        SIZE OF FCB WITHOUT LOGICALS
ID1LOGIC DS    128F              LOGICAL ROUTINES              IS
         ORG   ID1LOGIC+X'48'                                      950
ID1SAVE  DS    16F               SAVE AREA FOR P1-LOGICAL USE    950
         ORG
ID1LFCBS EQU   *-ID1LOGIC        SIZE OF LOGICAL PROCEDURE AREA
ID1FCBSZ EQU   *-ID1P1FST        SIZE OF P1 FCB
```

```
         DCACATE MF=D
         MFCHK SUPPORT=(C,D),PREFIX=D,MACID=CAC,MF=D,                  C
               DMACID=CAC,DNAME=CACATF
DCACATF  DSECT ,
               *,##### PREFIX=D, MACID=CAC #####
*
*---------------------------------------------------------------
* DECLARE FIXED PART OF CATALOG ENTRY
* ---------------------------------------------------------------
*
DCACFFP    DSECT
DCACFCEL    DS    XL2              LENGTH OF CATALOG ENTRY
*
DCACFCE#    EQU   4079             maximal length of catalog entry
DCACFEVE    DS    0XL2             VERSION OF CATALOG ENTRY LAYOUT
DCACFEV1    EQU   X'FF01',2
DCACFEV2    EQU   X'FF02',2          BS2000/OSD-BC V3
DCACFEV3    EQU   X'FF03',2          BS2000/OSD-BC V5               319
*
DCACFEFM    DS    X                  MARKER OF CE FORMAT
DCACFENW    EQU   X'FF'              "NEW" FORMAT INTRODUCED IN V1
*
DCACFVER    DS    X                  CE VERSION
DCACFVRC    EQU   X'03'              current CE version             319
DCACFEF1    EQU   X'01'                                             319
DCACFEF2    EQU   X'02'                                             319
DCACFEF3    EQU   X'03'                                             319
*
DCACFOMP    DS    XL2              OFFSET MAIN PART
*                                         JV ENTRY: MUST BE ZERO
DCACFOOP    DS    XL2              OFFSET OLD PART           IDJE/IDCE
DCACFOFJ    DS    XL2              OFFSET FILE-/JV-NAME
DCACFOEX    DS    XL2              OFFSET CE EXTENSION       IDCEX
*                                         JV ENTRY: MUST BE ZERO
DCACFOVE    DS    0XL2             OFFSET VOLUME EXTENT LIST IDVT, IDEE
DCACFOFI    DS    XL2                     OR FGG-INDEX  IDCEG
*                                         JV ENTRY: MUST BE ZERO
DCACFOCP    DS    XL2              OFFSET COMMON PART
DCACFODP    DS    XL2              OFFSET DATE PART
DCACFOUN    DS    XL2              OFFSET UNUSED
*                                         JV ENTRY: MUST BE ZERO
DCACFOHP    DS    XL2              OFFSET HSMS PART
*                                         JV ENTRY: MUST BE ZERO
DCACFOPP    DS    XL2              OFFSET PROFILE PART
*
DCACFOF#    EQU   (*-DCACFOMP)/L'DCACFOMP       number of offsets
            DS    (((*-DCACFFP+3)/4)*4+DCACFFP-*)XL1               310
DCACFFP#    EQU   *-DCACFFP
```

```
* ---------------------------------------------------------------
* HERE STARTS THE NEW VARIABLE PART OF THE CATALOG ENTRY
* (SEE DESCRIPTION FOR COMMON STRUCTURE, DATE STRUCTURE ETC.)
*
* THE OLD CE LAYOUT STARTS AFTER VARIABLE PART
* (SEE OLD IDCE, IDCEX ETC.)
* ---------------------------------------------------------------
*
*---------------------------------------------------------------
* DECLARE DATE STRUCTURE FOR FILE-/JV-ENTRY (UTC-Format)
* ---------------------------------------------------------------
*
DCACFDP     DSECT
DCACFFPL    DS    XL2             LENGTH OF DATE STRUCTURE FILE
DCACFJPL    DS    XL2             LENGTH OF DATE STRUCTURE JV
DCACFCRD    DS    OCL14           CREATION DATE PART
DCACFCRY    DS    CL4                  YEAR
DCACFCRM    DS    CL2                  MONTH
DCACFCRT    DS    CL2                  DAY
DCACFCRH    DS    CL2                  HOURS
DCACFCRI    DS    CL2                  MINUTES
DCACFCRS    DS    CL2                  SECONDS
*                                 DATE/TIME UNDEFINDED: C'        '
DCACFEXD    DS    OCL14           EXPIRATION DATE PART
DCACFEXY    DS    CL4                  YEAR
DCACFEXM    DS    CL2                  MONTH
DCACFEXT    DS    CL2                  DAY
DCACFEXH    DS    CL2                  HOURS
DCACFEXI    DS    CL2                  MINUTES
DCACFEXS    DS    CL2                  SECONDS
*                                 DATE/TIME UNDEFINDED: C'        '
DCACFLAD    DS    OCL14           LAST ACCESS DATE
DCACFLAY    DS    CL4                  YEAR
DCACFLAM    DS    CL2                  MONTH
DCACFLAT    DS    CL2                  DAY
DCACFLAH    DS    CL2                  HOURS
DCACFLAI    DS    CL2                  MINUTES
DCACFLAS    DS    CL2                  SECONDS
*                                 DATE/TIME UNDEFINDED: C'        '
DCACFLMD    DS    OCL14           LAST DATE MODIFIED
DCACFLMY    DS    CL4                  YEAR
DCACFLMM    DS    CL2                  MONTH
DCACFLMT    DS    CL2                  DAY
DCACFLMH    DS    CL2                  HOURS
DCACFLMI    DS    CL2                  MINUTES
DCACFLMS    DS    CL2                  SECONDS
*                                 DATE/TIME UNDEFINDED: C'        '
DCACFDED    DS    OCL14           DELETION DATE
```

```
DCACFDEY    DS    CL4                   YEAR
DCACFDEM    DS    CL2                   MONTH
DCACFDET    DS    CL2                   DAY
DCACFDEH    DS    CL2                   HOURS
DCACFDEI    DS    CL2                   MINUTES
DCACFDES    DS    CL2                   SECONDS
*                                 DATE/TIME UNDEFINDED: C'        '
            DS    XL2            RESERVED; MUST BE ZERO             310
            DS    (((*-DCACFDP+3)/4)*4+DCACFDP-*)XL1                310
DCACFDP#    EQU   *-DCACFDP
*
*-------------------------------------------------------------------
* DECLARE COMMON STRUCTURE FOR FILE/JV-ENTRY
*-------------------------------------------------------------------
*
DCACFCP     DSECT
DCACFCPL    DS    XL2                 LENGTH OF COMMON PART
DCACFEHC    DS    CL8                 EXTENDED HOST CODE
*           DC    C'          '         UNDEFINED
*           DC    C'DF04      '         EBCDIC.DF.04
DCACFCTS    DS    XL2                 CHARACTER TYPE SET
DCACFCUD    EQU   X'0000',2             UNDEFINED CHAR SET
DCACFEIN    DS    X                   INDICATOR
DCACFECC    EQU   X'80'                 CONVERTED CE
DCACFEUN    EQU   X'7F'                 UNUSED
DCACFCWT    DS    XL4                 TASKID OF LAST WRITE
DCACFCWS    DS    XL1                 SYSID OF LAST WRITE
DCACFWCC    DS    XL1                 WRITE CHECK COUNTER
* The following fields are present only if CE_IDENTIFICATION.VERSION
* in the fixed part of the CE is at least 2.
DCACFCOO    DS    XL5                 RESERVED: MUST BE ZERO
DCACFMGC    DS    CL8                 SMS MANAGEMENT CLASS
            DS    (((*-DCACFCP+3)/4)*4+DCACFCP-*)XL1                310
DCACFCP#    EQU   *-DCACFCP
*
            EJECT ,
*
* ------------------------------------------------------------------
* DECLARE NEW MAIN STRUCTURE
* ------------------------------------------------------------------
*
DCACFMP   DSECT
DCACFMPL   DS    XL2              LENGTH OF MAIN STRUCTURE
DCACFIND   DS    X                MAIN INDICATOR
DCACFPDF   EQU   X'80'                PD FILE
DCACFSNR   EQU   X'40'                SPACE NOT RELEASABLE
DCACFGUF   EQU   X'20'                GUARDS FILE
DCACFARW   EQU   X'10'                ARCHIVE WORK FILE
```

```
DCACFEX4   EQU   X'08'              EXTENT-LIST WITH 4 BYTE        319
DCACFSOW   EQU   X'04'              WORK FILE FOR SO MIGRATION
DCACFCMF   EQU   X'02'              CMS FILE
DCACFSRB   EQU   X'01'              SYSSRPM.BACKUP
DCACFACC   DS    XL4              ACCESS COUNTER
DCACFHCF   DS    0XL4             HSMS SAVED CODED FILE ID
DCACFTAP   DS    0X               TAPE CRITERION
DCACFBA#   DS    XL4              # BLOCKS OF TAPE FILE
DCACFBL#   DS    XL4              # BLOCKS RECORDED ON LAST TAPE
*                                   OF MULTIVOLUME TAPE FILE
DCACFPER   DS    X                PERFORMANCE ATTRIBUTE
DCACFPVH   EQU   X'03'               VERY HIGH
DCACFPHI   EQU   X'02'               HIGH
DCACFPST   EQU   X'01'               STANDARD    (DEFAULT VALUE)
DCACFUSE   DS    X                USE ATTRIBUTE
DCACFURW   EQU   X'03'               READ-WRITE  (DEFAULT VALUE)
DCACFUWR   EQU   X'02'               WRITE
DCACFURE   EQU   X'01'               READ
DCACFTPE   DS    X                PERFORMANCE ATTRIBUTE OPEN/CLOSE
DCACFTVH   EQU   X'03'               VERY HIGH
DCACFTHI   EQU   X'02'               HIGH
DCACFTST   EQU   X'01'               STANDARD    (DEFAULT VALUE)
DCACFTUS   DS    X                USE ATTRIBUTE OPEN/CLOSE
DCACFTRW   EQU   X'03'               READ-WRITE  (DEFAULT VALUE)
DCACFTWR   EQU   X'02'               WRITE
DCACFTRE   EQU   X'01'               READ
DCACFTYA   DS    X                DISK WRITE (FILE TYPE)
DCACFTCR   EQU   X'02'               BY CLOSE   (NOT SENSITIVE)
DCACFTNC   EQU   X'01'               IMMEDIATE  (SENSITIVE, DEFAULT)
DCACFCAI   DS    X                CACHE  INDICATOR
DCACFCAU   EQU   X'80'               S: CACHE USED
*                                    R: CACHE NOT AVAILBALE
DCACFDIG   EQU   X'40'               S: DATA IN GLOBAL STAORAGE
*                                    R: NO DATA IN GLOBAL STORAGE
DCACFDNO   EQU   X'20'               S: DENY OPEN REQUEST
*                                    R: OPEN POSSIBLE
DCACFDIC   EQU   X'10'               S: DATA STILL IN CACHE
*                                    R: DATA NO LONGER IN CACHE
DCACFWPF   EQU   X'08'               S: WRITE PERFORMED
*                                    R: WRITE NOT PERFORMED
DCACFFTS   DS    XL8              FILE TRANSFER INFORMATION
DCACFSIN   DS    X                SMS INDICATOR
DCACFAVA   EQU   X'C0'               AVAILABILITY:
DCACFAVS   EQU   X'00'                 STANDARD
DCACFAVH   EQU   X'40'                 HIGH
DCACFMIF   EQU   X'20'               S: SO MIGRATION FORBIDDEN
*                                    R: SO MIGRATION ALLOWED
DCACFSIR   EQU   X'1F'               RESERVED: MUST BE ZERO
```

```
DCACFSAT   DS    X               SMS FILE ATTRIBUTES
DCACFWRK   EQU   X'08'             S: FILE IS A USER WORKFILE
DCACFFMT   EQU   X'07'             FILE PRE-FORMAT:
DCACFFMU   EQU   X'00'               UNDEFINED
DCACFFMK   EQU   X'01'               K
DCACFFM2   EQU   X'02'               NK2
DCACFFM4   EQU   X'03'               NK4
DCACFIN1   DS    X               INDICATOR1
DCACFNER   EQU   X'80'             DO NOT ERASE
DCACFUSF   EQU   X'40'             USER-SPECIFIC SYSTEM FILE
DCACFD5R   EQU   X'20'             DELETE DURING F5 RECONSTRUCTION
DCACFCSF   EQU   X'10'             DELETE SPACEOPT H-FILES          318
DCACFNDA   EQU   X'08'             NO DMS ACCESS
DCACFIDF   EQU   X'04'             INCR/DECR FILE-VERSION-NR        319
DCACFI1R   EQU   X'03'             RESERVED
*
* The following fields are present only if CE_IDENTIFICATION.VERSION
* in the fixed part of the CE is at least 2.
DCACFUTU   DS    CL8             USER INFORMATION FOR STATE TU
DCACFUTP   DS    CL8             USER INFORMATION FOR STATE TPR
DCACFSTC   DS    CL8             STORAGE CLASS
DCACFTAG   DS    X               FILE TAG (IDENTIFIES SPECIFIC FILE)
DCACFTNO   EQU   0                 NONE
DCACFGCF   EQU   1                 GENERIC CATALOG
DCACFSIC   DS    X               SYSID OF WRITE CACHER            317
*
DCACFFS4   DS    XL4             FILE SIZE (4 BYTE !)             319
DCACFLP4   DS    XL4             LAST HALF PAGE POINTER (4 BYTE !) 319
           DS    XL2             RESERVED; MUST BE ZERO           319
           DS    (((*-DCACFMP+3)/4)*4+DCACFMP-*)XL1               310
DCACFMP#   EQU   *-DCACFMP      LENGTH OF STRUCTURE
*
*
* ---------------------------------------------------------------
* DECLARE NEW HSMS STRUCTURE FOR CATALOG ENTRY
* ---------------------------------------------------------------
*
DCACFHP    DSECT
DCACFHVS   DS    XL6              HSMS VOLUME SERIAL NUMBER
DCACFHVE   DS    XL1              VERSION OF HSMS STRUCTURE
DCACFHVL   DS    XL2              VARIABLE LENGTH OF HSMS STRUCTURE
DCACFHP#   EQU   *-DCACFHP
*
* ---------------------------------------------------------------
* DECLARE NEW PROFILE PART (VARIABLE IN LENGTH)
* ---------------------------------------------------------------
*
DCACFPP    DSECT
```

```
DCACFPPL    DS    XL2              LENGTH OF WHOLE PROFILE PART
DCACFPPV    DS    XL1              VERSION OF PROFILE PART
DCACFPPU    DS    XL4              :* Field has never been used
* ------------------------------------------------------------
* - HERE STARTS THE VARIABLE PART OF PROFILE NAME LIST
* - (THREE ENTRIES ARE DEFINED, 18 BYTES PER ENTRY)
* - PROFILE NAME: $USERID.NAME  OR  BLANK (MEANS UNDEFINED)
* ------------------------------------------------------------
DCACFPRD    DS    CL18             NAME OF READ PROFILE
DCACFPWR    DS    CL18             NAME OF WRITE PROFILE
DCACFPEX    DS    CL18             NAME OF EXEC PROFILE
            DS    XL3              ***     RESERVED        ***
DCACFPP#    EQU   *-DCACFPP
         EJECT ,
         IDCE  D,D,NEWONLY=*YES
*
* ------------------------------------------------------------
* - OLD CATALOG ENTRY FIXED PART
* ------------------------------------------------------------
         MFPRE DNAME=DCES,MF=D,ALIGN=X,PREFIX=D                    300
DDCES    DSECT ,
               *,##### PREFIX=D, MACID= #####
DDCEES   DS    XL2             0-1 ENTRY SIZE
DDCEPBN  DS    XL2             2-3 PRIMARY BLOCK NUMBER
DDCELBN  DS    XL2             4-5 BLOCK NUMBER
DDCEDMS  DS    XL1               6 DMS INDICATOR 1
DDCEINDX EQU   X'01'             S FILE GENERATION GROUP INDEX -V4.0-
DDCEEPP  EQU   X'02'                  S EXEC PASSWORD
DDCESF   EQU   X'04'             S SHARED FILE
DDCEROA  EQU   X'08'             S READ ONLY ACCESS
DDCERPP  EQU   X'10'             S READ PASSWORD
DDCEWPP  EQU   X'20'             S WRITE PASSWORD
DDCEAMN  EQU   X'C0'               PRINTER CONTROL
DDCEN    EQU   X'C0'             R NO CONTROL CHARACTER
DDCEM    EQU   X'40'             S MACHINE CODE CONTROL CHARACTER
DDCEA    EQU   X'C0'             S ASA CONTROL CHARACTER
DDCEDMS2 DS    XL1               7 DMS INDICATOR 2
*           EQU   X'80'              S USED BY FSTAT MACRO/CMD-NON SHARAB
*                                      (ONLY VERSIONS < V8.0)       V10.0
DDCEFAR  EQU   X'80'             S UDS - FAR INTERFACE          V10.0
DDCEFGEN EQU   X'40'             S FILE IS A GENERATION
DDCEFRAP EQU   X'20'             S FOREIGN RANDOM ACCESS PROCESSING
*                                   USED BY FILE MACRO AND CMS
DDCEISID EQU   X'10'             S ISAM INDEX AND DATA VOLUMES ARE
*                                   PRESENT IN CATALOG ENTRY
DDCERLK  EQU   X'08'             S: RELEASE LOCK
DDCEELK  EQU   X'04'             S: ERASE LOCK
DDCERPI  EQU   X'02'             S: FILE TO REPAIR INDICATOR (RPI)
```

```
                                                    15:29:06  2001-10-15   PAGE
0058
SOURCE STATEMENT
DDCEENCR EQU   X'01'              S: ENTRY WIYH ENCRYPTED PASSW
DDCERD   DS    CL4             8-11 READ PASSWORD
DDCEWRT  DS    CL4            12-15 WRITE PASSWORD
DDCESA   DS    XL2            16-17 SECONDARY ALLOCATION
DDCESAC  DS    XL1               18 SECONDARY ALLOCATION COUNT
DDCEEXTD DS    XL1               19 DISPLACEMENT TO CATALOG ENTRY
*                                   EXTENSION
         DS    CL6            UNDEFINDED DATA
DDCEMACL DS    XL1               26 MINI - ACL                    V10.0
DDCEURD  EQU   X'80'                S USER: READ-PRIVILEG        V10.0
DDCEUWR  EQU   X'40'                S USER: WRITE-PRIVILEG       V10.0
DDCEUEX  EQU   X'20'                S USER: EXEC-PRIVILEG        V10.0
DDCEGRD  EQU   X'10'                S GROUP: READ-PRIVILEG       V10.0
DDCEGWR  EQU   X'08'                S GROUP: WRITE-PRIVILEG      V10.0
DDCEGEX  EQU   X'04'                S GROUP: EXEC-PRIVILEG       V10.0
DDCEORD  EQU   X'02'                S OTHERS: READ-PRIVILEG      V10.0
DDCEOWR  EQU   X'01'                S OTHERS: WRITE-PRIVILEG     V10.0
DDCEDTVC DS    XL1               27 DATA ON TAPE VOLUME COUNT
DDCEFTP  DS    XL1               28 FCB FILE TYPE
DDCEFCBT EQU   X'C0'                FCB FILE TYPE
DDCESAM  EQU   X'C0'               R SAM FILE
DDCEISAM EQU   X'40'               S ISAM FILE
DDCEBTAM EQU   X'80'               S BTAM FILE
DDCEPAM  EQU   X'C0'               S PAM FILE
DDCERB   DS    XL1               29 RESTRICT BYTE
DDCEPRV  EQU   X'01'               S PRIVATE VOLUME
DDCEPUB  EQU   X'01'               R PUBLIC VOLUME
DDCERES  EQU   X'02'               S SYSRES RESTRICTED
                                                    15:29:06  2001-10-15   PAGE
0059
SOURCE STATEMENT
DDCEFOR  EQU   X'04'               S FOREIGN TAPE VOLUME
DDCEVIF  EQU   X'08'               S VERY IMPORTANT FILE ( VIF )
DDCEIDVC EQU   X'40'               S-IGNORE DATA VOLUME COUNT
*                                    FOR TAPE EOF PROCESSING
*        EQU   X'10'               S ERASE/CATAL COMMAND OR MACRO V10.0
*                                    IS ACTIVE                    V10.0
*                                    (ONLY VERSIONS < V8.0)       V10.0
*        EQU   X'20'               S OPEN/CLOSE IN PROCESS       V10.0
*                                    (ONLY VERSIONS < V9.0)       V10.0
DDCEDFB  EQU   X'80'               S File contains defective block
DDCEBLK  DS    XL2            30-31 BUFFER SIZE
DDCESTD  EQU   X'80'               S PAM CREATED FILE
DDCCODE        DS     XL1        32 CODE OF TAPE
DDCISO7        EQU    X'10'         ISO7 CODE
```

```
DDCEBCD         EQU    X'08'         EBCDIC CODE
DDCOWN          EQU    X'04'         OWN CODE ISO7
DDCNDEF         EQU    X'1C'         CODE NOT DEFINED
DDCERT    DS    XL1                33 RECORD TYPE
DDCEVFU   EQU   X'06'                 RECORD TYPE
DDCERTV   EQU   X'02'                 S VARIABLE
DDCERTF   EQU   X'04'                 S FIXED
DDCERTU   EQU   X'06'                 S UNDEFINED
DDCEREC   DS    XL2              34-35 RECORD LENGTH
DDCEKLC   DS    XL2              36-37 KEY LOCATION
DDCEKLH   DS    XL1                 38 KEY LENGTH
DDCEFSZ3  DS    XL3              39-41 FILE SIZE                          306
DDCELRP   DS    XL2              42-43 LAST RECORD POINTER
DDCEIRA   DS    XL1                 44 EDT VERSION-COUNTER
DDCELPP3  DS    XL3              45-47 LAST 1/2 PAGE POINTER              306
DDCEFN    DS    XL4              48-51 FILE NUMBER
DDCEVTD   DS    XL2              52-53 DISPLACEMENT TO THE LAST       020
*-                                        VSN TABLE ENTRY                 020
DDCEVN1   DS    XL1                 54 CURRENT VERSION NUMBER
DDCEVN2   DS    XL1                 55 BACK-UP VERSION NUMBER
DDCETYP   DS    XL1                 56 DEVICE TYPE
DDCEVC    DS    XL1                 57 VSN COUNT
DDCEF1    DS    XL1                 58 FORMAT 1 POINTER
DDCEFNC   DS    XL1                 59 FILENAME COUNT
DDCELEN   EQU   *-DDCEES               LENGTH OF FIXED PORTION
DDCEFNM   DS    CL1           60-VAR FILENAME
*-
DDCEESF   EQU   1881          MAXIMAL SIZE OF EXTENTLIST WITH 3 BYTE
*-                            OR F1LABEL                               307
DDCEES4   EQU   2496          MAXIMAL SIZE OF EXTENTLIST WITH 4 BYTE   306
          EJECT ,
          IDCEX D,D,NEWONLY=*YES
*
*************************************************************************
***            CATALOG ENTRY EXTENSION
*************************************************************************
          MFPRE DNAME=DCEXS,MF=D,ALIGN=X,PREFIX=D                        300
DDCEXS    DSECT ,
                *,##### PREFIX=D, MACID= #####
*                                                                        CEL
DDCEEXTS  EQU   *                     START OF CATALOG ENTRY EXTENSION
DDCECNT   DS    XL2                   FILE RECONSTRUCTION COUNT          CEL
          ORG   DDCECNT                                                  CEL
DDCESESN  DS    XL1       SESSION NUMBER                               CEL
DDCEOCI   DS    XL1       OPEN-CLOSE INDICATOR -- OCI                  CEL
DDCECMSI  DS    XL1                   CMS INDICATOR FLAGS              CEL
DDCEWRCH  EQU   X'C0'                 7-6 WRITE COUNT (MODULO 4)       CEL
DDCEINCR  EQU   X'40'                 COUNT INCREMENT                  CEL
```

```
DDCEEXCL EQU   X'20'                  5 S=EXCLUSIVE SECURE          CEL
DDCESVAC EQU   X'10'                  4 S=ACCESSIBLE FOR SERVICE    CEL
DDCEKEYZ EQU   X'0C'                  3-2 S: BLKCTRL=NONE           9.5
*                                         R: BLKCTRL=PAMKEY         9.5
DDCEKEYD EQU   X'08'                  3   S: BLKCTRL=NO             9.5
DDCEKEYV EQU   X'04'                  2   S: BLKCTRL=DATA           9.5
DDCESPCE EQU   X'02'                  1 RESERVED FOR SPANNED CE'S   CEL
DDCESPC1 EQU   X'01'                  0 RESERVED FOR SPANNED CE'S   CEL
*                                                                   CEL
         DS    XL1                   OPEN COUNTER                  V10.0
*                                    (ONLY VERSIONS < V9.0)        V10.0
DDCEDMS3 DS    X                    DMS INDICATOR 3                 CEL
DDCEPWSK EQU   X'80'                SKIP PASSWORDS                  CEL
DDCEPSCL EQU   X'40'                SET: PSEUDO - CLOSE             CEL
DDCETEMP EQU   X'20'                SET: TEMPORARY INDICATOR        CEL
DDCENIMC EQU   X'10'                SET: NOT TO BE ERASED BY IMCAT  CEL
DDCESPEC EQU   X'08'                SET: PLAM-FILE INDICATOR        9.5
DDCESUSA EQU   X'04'                SPEC_USER_SPACE_ACCG            CEL
                                                 15:29:06  2001-10-15   PAGE
0062
SOURCE STATEMENT
DDCEOEX  EQU   X'02'                S: MINI-ACL: OTHERS: EXEC-PRIV. V10.0
DDCEFACP EQU   X'01'                S: FULL ACL PRESENT             032
DDCAUDIT DC    X'00'                    AUDIT KEY                  CEL
DDCAUDNO EQU   X'00'                    AUDIT=NONE                 CEL
DDCAUDSU EQU   X'01'                    AUDIT=SUCC                 CEL
DDCAUDFA EQU   X'02'                    AUDIT=FAIL                 CEL
DDCAUDAL EQU   X'04'                    AUDIT=ALL                  CEL
DDCEFRS  DS    XL1          FILE RECONSTRUCTION SYSTEM CONTROL      CEL
*                                 BYTE                            CEL
DDCEFRRC EQU   X'80'         S RECON=YES                            CEL
DDCEFRRS EQU   X'40'         S RESET=YES                            CEL
DDCEOPT  EQU   X'20'         S CANNOT OPEN FILE OUTPUT/OUTIN        CEL
DDCECHNG EQU   X'10'         S CALL FILE RECONSTRUCTION SYSTEM      CEL
*                                    BEFORE OPEN                  CEL
DDCEMODE EQU   X'08'         S MOVE MODE ONLY                       CEL
DDCERLSK EQU   X'04'         S RESET FUNCTION PENDING--FILE         CEL
*                                    CANNOT BE OPENED             CEL
DDCMIGFB EQU   3             MIGRATION FORBIDDEN
DDCERESS EQU   2             MIGRATION INHIBIT
DDCETRES EQU   1             MIGRATION LIMITED
DDCEIIND DS    XL1                FILE INDICATOR BYTE            CEL
DDCEMACP EQU   X'80'               S MINI ACL PRESENT             032
DDCESUW1 EQU   X'40'               SHARED UPDATE WEAK             100
DDCESUW2 EQU   X'20'               SHARED UPDATE WEAK
DDCESUW3 EQU   X'10'               SHARED UPDATE WEAK
DDCESUW4 EQU   X'08'               SHARED UPDATE WEAK
DDCENKI4 EQU   X'02'               NK-ISAM 4K-BLOCK-FILE          100
```

```
DDCEVMIN EQU   X'01'              R MIN FUNCTION FOR ISAM VALUE FLAG
DDCEVMAX EQU   X'01'              S MAX FUNCTION FOR ISAM VALUE FLAG
DDCELOGL DS    XL1                ISAM LOGICAL FLAG LENGTH         CEL
DDCEVALL DS    XL1                ISAM VALUE FLAG LENGTH           CEL
DDCESAD  DS    XL2                SECONDARY ALLOCATION(DATA)       CEL
DDCEITN8 DS    XL1                ITN OF LOCKER WITH VERSION # < 9 CEL
DDCELOCK DS    XL1                LOCKS                            CEL
*                                 XXXX.... OUTPUT LOCKS            CEL
*                                 ....XXXX CATALOG LOCKS           CEL
*                                                                  CEL
DDCE$UID DS    CL8                ALPHA-NUMERIC $USERID            CEL
         DS    XL4           UNDEFINDED DATA
DDCEEX   DS    XL4                    EXEC PASSWORD                CEL
DDCEF1EX DS    XL1       LEFTMOST BYTE OF F1 POINTER               CEL
*                                                                  CEL
DDCEMEMB DS    4CL2      SHARER-ID/INPUT COUNTER (4 TIMES)         CEL
* SHARER-1 : SHARER-ID : 1 BYTE ; INPUT COUNTER : 1 BYTE           CEL
* SHARER-2 : SAME --> 2 BYTES                                      CEL
* SHARER-3 : SAME --> 2 BYTES                                      CEL
* SHARER-4 : SAME --> 2 BYTES                                      CEL
         ORG   DDCEMEMB                                           CEL
* THE BYTES USED BY THE PRIVATE DISK FILES ARE REDEFINED FOR USE BY CEL
* TAPE FILES.                                                     CEL
DDCETPID DS    X                  TAPE INDICATORS                  CEL
DDCEPFT  EQU   X'80'              S - PERMANENT FOREIGN TAPE       CEL
DDCESTDL EQU   X'40'              S - STD LABELS                   CEL
DDCENSTD EQU   X'20'              S -NSTD LABELS                   CEL
DDCENOLB EQU   X'10'              S -NO LABELS                     CEL
DDCEDIN3 EQU   X'08'              S -LABEL TYPE=DIN3               CEL
DDCEDIN2 EQU   X'04'              S -           DIN2               CEL
DDCEDIN1 EQU   X'02'              S -           DIN1               CEL
DDCEBS2  EQU   X'01'              S -           BS2000             CEL
DDCEBUFO DS    X                  BUFFER OFFSET                    CEL
DDCEFSEQ DS    XL2                FILE SEQUENCE NUMBER             CEL
DDCEDLAB DS    X                  DIN LABEL                        100
DDCEDL00 EQU   X'00'              DIN LABEL UNDEFINED              100
         DS    CL3                UNUSED                           CEL
*                                                                  CEL
*                                                                  CEL
DDCEARLV DC    X'00'                                              CEL
* THE FIRST 4 BITS ARE RESERVED FOR SAVE/RESTORE                   CEL
DDCEARCA EQU   X'00'    BACKUP LEVEL=A                             CEL
DDCEARCB EQU   X'10'    BACKUP LEVEL=B                             CEL
DDCEARCC EQU   X'20'    BACKUP LEVEL=C                             CEL
DDCEARCD EQU   X'30'    BACKUP LEVEL=D                             CEL
DDCEARCE EQU   X'40'    BACKUP LEVEL=E                             CEL
*                                                                  CEL
         ORG   DDCEARLV                                           CEL
```

```
DDCEARFL DC    X'00'    FLAGS                                          CEL
DDCELARG EQU   X'01'    LARGE FILE                                     CEL
DDCEDEST EQU   X'02'    TO BE ERASED WITH DESTROY                      CEL
DDCEPART EQU   X'04'    PART RESTORE                                   CEL
DDCEMIGR EQU   X'08'    S: MIGRATED (HSMS)                             CEL
*                       R: NOT MIGRATED (HSMS)                         CEL
*                                                                      CEL
*                                                                      CEL
DDCEEXTL EQU   *-DDCEEXTS          LENGTH OF CATALOG ENTRY EXTENSION
         EJECT ,
         IDVT  D,D
*
* ------------------------------------------------------------
* - VOLUME TABLE ENTRY
* ------------------------------------------------------------
         MFPRE DNAME=DVTS,MF=D,ALIGN=X,PREFIX=D                        300
DDVTS    DSECT ,
               *,##### PREFIX=D, MACID= #####
DDVTVSN  DS    CL6             0-5 VOLUME SERIAL NUMBER
DDVTLENT EQU   *-DDVTVSN             LENGTH OF FIXED PORTION-TAPE
DDVTTYP  DS    XL1               6 DEVICE TYPE
DDVTIDV  EQU   X'40'               6-6 S ISAM DATA VOLUME
DDVTIIV  EQU   X'40'               6-6 R ISAM INDEX VOLUME
DDVT#EE  DS    XL1               7 NUMBER OF EXTENT ENTRIES FOR VSN
DDVTLEN  EQU   *-DDVTVSN             LENGTH OF FIXED PORTION-R.A.
         EJECT ,
         IDEE  D,D,NEWONLY=*YES                                        319
*
* ------------------------------------------------------------
* - CATALOG ENTRY EXTENT LIST ENTRY WITH 3 BYTE
* ------------------------------------------------------------
         MFPRE DNAME=DEE3,MF=D,ALIGN=X,PREFIX=D                        301
DDEE3    DSECT ,
               *,##### PREFIX=D, MACID= #####
*
DDEELHP3 DS    XL3                    LOGICAL 1/2 PG                 301
DDEEPHP3 DS    XL3                    PHYSICAL 1/2 PG                301
*
DDEELEN3 EQU   *-DDEELHP3             LENGTH OF ENTRY                301
         EJECT ,
                                                15:29:06  2001-10-15   PAGE
0067
SOURCE STATEMENT
         DCAEE4 MF=D,PREFIX=D                                          319
         MFTST MF=D,PREFIX=D,MACID=CAE,ALIGN=X,                           C
               DMACID=CAE,SUPPORT=(D,C),DNAME=CAEE4
DCAEE4   DSECT ,
               *,##### PREFIX=D, MACID=CAE #####
```

```
*   CATALOG ENTRY EXTENT LIST ENTRY WITH 4 BYTE
DCAELHP4 DS    XL4                      4 BYTE LOGICAL HALF-PAGE
DCAEPHP4 DS    XL4                      4 BYTE PHYSICAL HALF-PAGE
DCAE#    EQU   *-DCAELHP4
         EJECT ,
         IDCEG D,D
*
* ---------------------------------------------------------------
* - CATALOG ENTRY EXTENSION FOR FILE GENERATION GROUP
* - THIS ENTRY IS LINKED AFTER CATALOG ENTRY EXTENSION (IDCEX)
* - (INSTEAD OF VOLUME TABLE (IDVT))
* ---------------------------------------------------------------
         MFPRE DNAME=DCEG,MF=D,ALIGN=X,PREFIX=D                    300
DDCEG    DSECT ,
              *,##### PREFIX=D, MACID= #####
DDCEGBAS    DS    XL2        BASE FOR RELATIVE ADDRESSING
DDCEGCUR    DS    XL2        CURRENT GENERATION
DDCEGBEG    DS    XL2        FIRST GENERATION
DDCEGGEN    DS    XL2        GEN=
DDCEGDIS    DS    X          DISP=
DDCEGCYC    EQU   X'01'          =CYCLE
DDCEGDEL    EQU   X'02'          =DELETE
DDCEGREU    EQU   X'03'          =REUSE
DDCEGKEE    EQU   X'04'          =KEEP                              700
DDCEGDEV    DS    X          DEVICE TYPR (PR.DISK)                 003
DDCEGVSN    DS    CL6        VOLUME #    (PR.DISK)                 003
DDCEGNUS    DS    CL5        NOT USED                              003
DDCEGLEN    EQU *-DDCEGBAS
         IDCE D,D,NEWONLY=*YES
*
* -------------------------------------------------------
* - OLD CATALOG ENTRY FIXED PART
* -------------------------------------------------------
         MFPRE DNAME=DCES,MF=D,ALIGN=X,PREFIX=D                    300
DDCES    DSECT ,
              *,##### PREFIX=D, MACID= #####
DDCEES   DS    XL2             0-1 ENTRY SIZE
DDCEPBN  DS    XL2             2-3 PRIMARY BLOCK NUMBER
DDCELBN  DS    XL2             4-5 BLOCK NUMBER
DDCEDMS  DS    XL1               6 DMS INDICATOR 1
DDCEINDX EQU   X'01'             S FILE GENERATION GROUP INDEX -V4.0-
DDCEEPP  EQU   X'02'                   S EXEC PASSWORD
                                              15:29:06  2001-10-15   PAGE
0069
SOURCE STATEMENT
DDCESF   EQU   X'04'             S SHARED FILE
DDCEROA  EQU   X'08'             S READ ONLY ACCESS
DDCERPP  EQU   X'10'             S READ PASSWORD
```

```
DDCEWPP  EQU   X'20'            S WRITE PASSWORD
DDCEAMN  EQU   X'C0'              PRINTER CONTROL
DDCEN    EQU   X'C0'            R NO CONTROL CHARACTER
DDCEM    EQU   X'40'            S MACHINE CODE CONTROL CHARACTER
DDCEA    EQU   X'C0'            S ASA CONTROL CHARACTER
DDCEDMS2 DS    XL1              7 DMS INDICATOR 2
*           EQU   X'80'             S USED BY FSTAT MACRO/CMD-NON SHARAB
*                                     (ONLY VERSIONS < V8.0)       V10.0
DDCEFAR  EQU   X'80'            S UDS - FAR INTERFACE          V10.0
DDCEFGEN EQU   X'40'            S FILE IS A GENERATION
DDCEFRAP EQU   X'20'            S FOREIGN RANDOM ACCESS PROCESSING
*                                  USED BY FILE MACRO AND CMS
DDCEISID EQU   X'10'            S ISAM INDEX AND DATA VOLUMES ARE
*                                  PRESENT IN CATALOG ENTRY
DDCERLK  EQU   X'08'            S: RELEASE LOCK
DDCEELK  EQU   X'04'            S: ERASE LOCK
DDCERPI  EQU   X'02'            S: FILE TO REPAIR INDICATOR (RPI)
DDCEENCR EQU   X'01'            S: ENTRY WIYH ENCRYPTED PASSW
DDCERD   DS    CL4           8-11 READ PASSWORD
DDCEWRT  DS    CL4          12-15 WRITE PASSWORD
DDCESA   DS    XL2          16-17 SECONDARY ALLOCATION
DDCESAC  DS    XL1             18 SECONDARY ALLOCATION COUNT
DDCEEXTD DS    XL1             19 DISPLACEMENT TO CATALOG ENTRY
*                                 EXTENSION
         DS    CL6          UNDEFINDED DATA
DDCEMACL DS    XL1             26 MINI - ACL                  V10.0
DDCEURD  EQU   X'80'               S USER: READ-PRIVILEG      V10.0
DDCEUWR  EQU   X'40'               S USER: WRITE-PRIVILEG     V10.0
DDCEUEX  EQU   X'20'               S USER: EXEC-PRIVILEG      V10.0
DDCEGRD  EQU   X'10'               S GROUP: READ-PRIVILEG     V10.0
DDCEGWR  EQU   X'08'               S GROUP: WRITE-PRIVILEG    V10.0
DDCEGEX  EQU   X'04'               S GROUP: EXEC-PRIVILEG     V10.0
DDCEORD  EQU   X'02'               S OTHERS: READ-PRIVILEG    V10.0
DDCEOWR  EQU   X'01'               S OTHERS: WRITE-PRIVILEG   V10.0
DDCEDTVC DS    XL1             27 DATA ON TAPE VOLUME COUNT
DDCEFTP  DS    XL1             28 FCB FILE TYPE
DDCEFCBT EQU   X'C0'               FCB FILE TYPE
DDCESAM  EQU   X'C0'             R SAM FILE
DDCEISAM EQU   X'40'             S ISAM FILE
DDCEBTAM EQU   X'80'             S BTAM FILE
DDCEPAM  EQU   X'C0'             S PAM FILE
DDCERB   DS    XL1             29 RESTRICT BYTE
DDCEPRV  EQU   X'01'             S PRIVATE VOLUME
DDCEPUB  EQU   X'01'             R PUBLIC VOLUME
DDCERES  EQU   X'02'             S SYSRES RESTRICTED
DDCEFOR  EQU   X'04'             S FOREIGN TAPE VOLUME
DDCEVIF  EQU   X'08'             S VERY IMPORTANT FILE ( VIF )
DDCEIDVC EQU   X'40'             S-IGNORE DATA VOLUME COUNT
```

```
*                                    FOR TAPE EOF PROCESSING
*        EQU   X'10'             S ERASE/CATAL COMMAND OR MACRO V10.0
*                                    IS ACTIVE                  V10.0
*                                    (ONLY VERSIONS < V8.0)     V10.0
*        EQU   X'20'             S OPEN/CLOSE IN PROCESS        V10.0
*                                    (ONLY VERSIONS < V9.0)     V10.0
DDCEDFB  EQU   X'80'             S File contains defective block
DDCEBLK  DS    XL2           30-31 BUFFER SIZE
DDCESTD  EQU   X'80'             S PAM CREATED FILE
DDCCODE        DS     XL1       32 CODE OF TAPE
DDCISO7        EQU    X'10'        ISO7 CODE
DDCEBCD        EQU    X'08'        EBCDIC CODE
DDCOWN         EQU    X'04'        OWN CODE ISO7
DDCNDEF        EQU    X'1C'        CODE NOT DEFINED
DDCERT   DS    XL1              33 RECORD TYPE
DDCEVFU  EQU   X'06'                RECORD TYPE
DDCERTV  EQU   X'02'              S VARIABLE
DDCERTF  EQU   X'04'              S FIXED
DDCERTU  EQU   X'06'              S UNDEFINED
DDCEREC  DS    XL2           34-35 RECORD LENGTH
DDCEKLC  DS    XL2           36-37 KEY LOCATION
DDCEKLH  DS    XL1              38 KEY LENGTH
DDCEFSZ3 DS    XL3           39-41 FILE SIZE                      306
DDCELRP  DS    XL2           42-43 LAST RECORD POINTER
DDCEIRA  DS    XL1              44 EDT VERSION-COUNTER
DDCELPP3 DS    XL3           45-47 LAST 1/2 PAGE POINTER          306
DDCEFN   DS    XL4           48-51 FILE NUMBER
DDCEVTD  DS    XL2           52-53 DISPLACEMENT TO THE LAST      020
*-                                     VSN TABLE ENTRY             020
DDCEVN1  DS    XL1              54 CURRENT VERSION NUMBER
DDCEVN2  DS    XL1              55 BACK-UP VERSION NUMBER
DDCETYP  DS    XL1              56 DEVICE TYPE
DDCEVC   DS    XL1              57 VSN COUNT
DDCEF1   DS    XL1              58 FORMAT 1 POINTER
DDCEFNC  DS    XL1              59 FILENAME COUNT
DDCELEN  EQU   *-DDCEES               LENGTH OF FIXED PORTION
DDCEFNM  DS    CL1           60-VAR FILENAME
*-
DDCEESF  EQU   1881          MAXIMAL SIZE OF EXTENTLIST WITH 3 BYTE
*-                           OR F1LABEL                           307
DDCEES4  EQU   2496          MAXIMAL SIZE OF EXTENTLIST WITH 4 BYTE  306
         IDCEX D,D,NEWONLY=*YES
*
*********************************************************************
***            CATALOG ENTRY EXTENSION
*********************************************************************
         MFPRE DNAME=DCEXS,MF=D,ALIGN=X,PREFIX=D                    300
DDCEXS   DSECT ,
```

```
                *,##### PREFIX=D, MACID= #####
*                                                                      CEL
DDCEEXTS EQU    *                   START OF CATALOG ENTRY EXTENSION
DDCECNT  DS     XL2                 FILE RECONSTRUCTION COUNT          CEL
         ORG    DDCECNT                                                CEL
DDCESESN DS     XL1         SESSION NUMBER                             CEL
DDCEOCI  DS     XL1         OPEN-CLOSE INDICATOR -- OCI                CEL
DDCECMSI DS     XL1                 CMS INDICATOR FLAGS                CEL
DDCEWRCH EQU    X'C0'               7-6 WRITE COUNT (MODULO 4)         CEL
DDCEINCR EQU    X'40'               COUNT INCREMENT                    CEL
DDCEEXCL EQU    X'20'                   5 S=EXCLUSIVE SECURE           CEL
DDCESVAC EQU    X'10'                   4 S=ACCESSIBLE FOR SERVICE     CEL
DDCEKEYZ EQU    X'0C'                   3-2 S: BLKCTRL=NONE            9.5
*                                           R: BLKCTRL=PAMKEY          9.5
DDCEKEYD EQU    X'08'                   3   S: BLKCTRL=NO              9.5
DDCEKEYV EQU    X'04'                   2   S: BLKCTRL=DATA            9.5
DDCESPCE EQU    X'02'                   1 RESERVED FOR SPANNED CE'S    CEL
DDCESPC1 EQU    X'01'                   0 RESERVED FOR SPANNED CE'S    CEL
*                                                                      CEL
         DS     XL1                 OPEN COUNTER                     V10.0
*                                   (ONLY VERSIONS < V9.0)           V10.0
DDCEDMS3 DS     X                  DMS INDICATOR 3                     CEL
DDCEPWSK EQU    X'80'              SKIP PASSWORDS                      CEL
DDCEPSCL EQU    X'40'              SET: PSEUDO - CLOSE                 CEL
DDCETEMP EQU    X'20'              SET: TEMPORARY INDICATOR            CEL
DDCENIMC EQU    X'10'              SET: NOT TO BE ERASED BY IMCAT      CEL
DDCESPEC EQU    X'08'              SET: PLAM-FILE INDICATOR            9.5
DDCESUSA EQU    X'04'              SPEC_USER_SPACE_ACCG                CEL
DDCEOEX  EQU    X'02'              S: MINI-ACL: OTHERS: EXEC-PRIV.  V10.0
DDCEFACP EQU    X'01'              S: FULL ACL PRESENT                 032
DDCAUDIT DC     X'00'                  AUDIT KEY                       CEL
DDCAUDNO EQU    X'00'                  AUDIT=NONE                      CEL
DDCAUDSU EQU    X'01'                  AUDIT=SUCC                      CEL
DDCAUDFA EQU    X'02'                  AUDIT=FAIL                      CEL
DDCAUDAL EQU    X'04'                  AUDIT=ALL                       CEL
DDCEFRS  DS     XL1         FILE RECONSTRUCTION SYSTEM CONTROL         CEL
*                                 BYTE                                 CEL
DDCEFRRC EQU    X'80'       S RECON=YES                                CEL
DDCEFRRS EQU    X'40'       S RESET=YES                                CEL
DDCEOPT  EQU    X'20'       S CANNOT OPEN FILE OUTPUT/OUTIN            CEL
DDCECHNG EQU    X'10'       S CALL FILE RECONSTRUCTION SYSTEM          CEL
*                                   BEFORE OPEN                       CEL
DDCEMODE EQU    X'08'       S MOVE MODE ONLY                           CEL
DDCERLSK EQU    X'04'       S RESET FUNCTION PENDING--FILE             CEL
*                                   CANNOT BE OPENED                  CEL
DDCMIGFB EQU    3           MIGRATION FORBIDDEN
DDCERESS EQU    2           MIGRATION INHIBIT
DDCETRES EQU    1           MIGRATION LIMITED
```

```
DDCEIIND DS    XL1                 FILE INDICATOR BYTE              CEL
DDCEMACP EQU   X'80'                S MINI ACL PRESENT              032
DDCESUW1 EQU   X'40'                SHARED UPDATE WEAK              100
DDCESUW2 EQU   X'20'                SHARED UPDATE WEAK
DDCESUW3 EQU   X'10'                SHARED UPDATE WEAK
DDCESUW4 EQU   X'08'                SHARED UPDATE WEAK
DDCENKI4 EQU   X'02'                NK-ISAM 4K-BLOCK-FILE           100
DDCEVMIN EQU   X'01'                R MIN FUNCTION FOR ISAM VALUE FLAG
DDCEVMAX EQU   X'01'                S MAX FUNCTION FOR ISAM VALUE FLAG
DDCELOGL DS    XL1                 ISAM LOGICAL FLAG LENGTH         CEL
DDCEVALL DS    XL1                 ISAM VALUE FLAG LENGTH           CEL
DDCESAD  DS    XL2                 SECONDARY ALLOCATION(DATA)       CEL
DDCEITN8 DS    XL1                 ITN OF LOCKER WITH VERSION # < 9 CEL
DDCELOCK DS    XL1                 LOCKS                            CEL
*                                  XXXX.... OUTPUT LOCKS            CEL
*                                  ....XXXX CATALOG LOCKS           CEL
*                                                                   CEL
DDCE$UID DS    CL8                 ALPHA-NUMERIC $USERID            CEL
         DS    XL4           UNDEFINDED DATA
DDCEEX   DS    XL4                     EXEC PASSWORD                CEL
DDCEF1EX DS    XL1       LEFTMOST BYTE OF F1 POINTER                CEL
*                                                                   CEL
DDCEMEMB DS    4CL2      SHARER-ID/INPUT COUNTER (4 TIMES)          CEL
* SHARER-1 : SHARER-ID : 1 BYTE ; INPUT COUNTER : 1 BYTE            CEL
* SHARER-2 : SAME --> 2 BYTES                                       CEL
* SHARER-3 : SAME --> 2 BYTES                                       CEL
* SHARER-4 : SAME --> 2 BYTES                                       CEL
         ORG   DDCEMEMB                                            CEL
* THE BYTES USED BY THE PRIVATE DISK FILES ARE REDEFINED FOR USE BY CEL
* TAPE FILES.                                                       CEL
DDCETPID DS    X                   TAPE INDICATORS                  CEL
DDCEPFT  EQU   X'80'                S - PERMANENT FOREIGN TAPE       CEL
DDCESTDL EQU   X'40'                S - STD LABELS                   CEL
DDCENSTD EQU   X'20'                S -NSTD LABELS                   CEL
DDCENOLB EQU   X'10'                S -NO LABELS                     CEL
DDCEDIN3 EQU   X'08'                S -LABEL TYPE=DIN3               CEL
DDCEDIN2 EQU   X'04'                S -           DIN2               CEL
DDCEDIN1 EQU   X'02'                S -           DIN1               CEL
DDCEBS2  EQU   X'01'                S -           BS2000             CEL
DDCEBUFO DS    X                   BUFFER OFFSET                    CEL
DDCEFSEQ DS    XL2                 FILE SEQUENCE NUMBER             CEL
DDCEDLAB DS    X                   DIN LABEL                        100
DDCEDLOO EQU   X'00'                DIN LABEL UNDEFINED             100
         DS    CL3                 UNUSED                           CEL
*                                                                   CEL
*                                                                   CEL
DDCEARLV DC    X'00'                                                CEL
* THE FIRST 4 BITS ARE RESERVED FOR SAVE/RESTORE                    CEL
```

```
DDCEARCA EQU   X'00'    BACKUP LEVEL=A                                  CEL
DDCEARCB EQU   X'10'    BACKUP LEVEL=B                                  CEL
DDCEARCC EQU   X'20'    BACKUP LEVEL=C                                  CEL
DDCEARCD EQU   X'30'    BACKUP LEVEL=D                                  CEL
DDCEARCE EQU   X'40'    BACKUP LEVEL=E                                  CEL
*                                                                       CEL
         ORG   DDCEARLV                                                 CEL
DDCEARFL DC    X'00'    FLAGS                                           CEL
DDCELARG EQU   X'01'    LARGE FILE                                      CEL
DDCEDEST EQU   X'02'    TO BE ERASED WITH DESTROY                       CEL
DDCEPART EQU   X'04'    PART RESTORE                                    CEL
DDCEMIGR EQU   X'08'    S: MIGRATED (HSMS)                              CEL
*                       R: NOT MIGRATED (HSMS)                          CEL
*                                                                       CEL
*                                                                       CEL
DDCEEXTL EQU   *-DDCEEXTS          LENGTH OF CATALOG ENTRY EXTENSION
         IDVT D,D
*
* ------------------------------------------------------------
* - VOLUME TABLE ENTRY
* ------------------------------------------------------------
         MFPRE DNAME=DVTS,MF=D,ALIGN=X,PREFIX=D                         300
DDVTS    DSECT ,
               *,##### PREFIX=D, MACID= #####
DDVTVSN  DS    CL6             0-5 VOLUME SERIAL NUMBER
DDVTLENT EQU   *-DDVTVSN           LENGTH OF FIXED PORTION-TAPE
DDVTTYP  DS    XL1               6 DEVICE TYPE
DDVTIDV  EQU   X'40'               6-6 S ISAM DATA VOLUME
DDVTIIV  EQU   X'40'               6-6 R ISAM INDEX VOLUME
DDVT#EE  DS    XL1               7 NUMBER OF EXTENT ENTRIES FOR VSN
DDVTLEN  EQU   *-DDVTVSN           LENGTH OF FIXED PORTION-R.A.
         IDEE D,D
*
* ------------------------------------------------------------
* - CATALOG ENTRY EXTENT LIST ENTRY WITH 3 BYTE
* ------------------------------------------------------------
         MFPRE DNAME=DEES,MF=D,ALIGN=X,PREFIX=D                         300
DDEES    DSECT ,
               *,##### PREFIX=D, MACID= #####
*
DDEELHP  DS    XL3                     LOGICAL 1/2 PG
DDEEPHP  DS    XL3                     PHYSICAL 1/2 PG
*
DDEELEN  EQU   *-DDEELHP               LENGTH OF ENTRY
         IDCEG D,D
```

```
*
* ------------------------------------------------------------
* - CATALOG ENTRY EXTENSION FOR FILE GENERATION GROUP
* - THIS ENTRY IS LINKED AFTER CATALOG ENTRY EXTENSION (IDCEX)
* - (INSTEAD OF VOLUME TABLE (IDVT))
* ------------------------------------------------------------
         MFPRE DNAME=DCEG,MF=D,ALIGN=X,PREFIX=D                    300
DDCEG    DSECT ,
               *,##### PREFIX=D, MACID= #####
DDCEGBAS    DS    XL2        BASE FOR RELATIVE ADDRESSING
DDCEGCUR    DS    XL2        CURRENT GENERATION
DDCEGBEG    DS    XL2        FIRST GENERATION
DDCEGGEN    DS    XL2        GEN=
DDCEGDIS    DS    X          DISP=
DDCEGCYC    EQU   X'01'          =CYCLE
DDCEGDEL    EQU   X'02'          =DELETE
DDCEGREU    EQU   X'03'          =REUSE
DDCEGKEE    EQU   X'04'          =KEEP                             700
DDCEGDEV    DS    X          DEVICE TYPR (PR.DISK)               003
DDCEGVSN    DS    CL6        VOLUME #    (PR.DISK)               003
DDCEGNUS    DS    CL5        NOT USED                            003
DDCEGLEN    EQU *-DDCEGBAS
```

```
         $JTBP MF=D
         MFTST MF=D,PREFIX=E,MACID=JT,ALIGN=F,                          C
               DMACID=JT,SUPPORT=(D,C),DNAME=JTBP
EJTBP    DSECT ,
               *,##### PREFIX=E, MACID=JT #####
EJTPLEN              EQU   608
*
EJTPIDNT             DS    CL4           String 'JTBP' identifies
*                                        itself
EJTPVERS             DS    CL3           BS2000-Version that created
*                                        this JTBP
EJTPJOMT             DS    FL1           indicates which structure of
*                                        union 'miscellaneous' (see
*                                        below) is valid
*   indicates which of the structures in union 'miscellaneous' is
*   valid
EJTPSMUN             EQU   0             X'00'
EJTPSMST             EQU   40            ' ': start
EJTPSMCA             EQU   195           'C': calendar
EJTPSMRP             EQU   217           'R': repeat
EJTPSMSU             EQU   226           'S': subtask
EJTRBP               EQU   231           'X': remote batch
*
*
EJTPJOBB             DS    0XL44         JOBBAS: basic job data
EJTPUSR              DS    CL8           (USERID) user id
EJTPACT              DS    CL8           (ACT#) account number
EJTPPSWD             DS    CL8           password of user Id - only if
*                                        invalid! (for use in
*                                        exit#030)
EJTPJCLS             DS    CL8           job class
EJTJNAM              DS    CL8           (JOBNAME) job name
EJTPTSN              DS    CL4           TSN
*
*
EJTPST               DS    0XL12         STATE: data on job state
EJTPSTCI             DS    A             (CANCEL_INFO@) pointer to
*                                        cancel context
EJTPSTST             DS    F             time the job started
EJTPSTQU             DS    X             (QUE#) queue number
EJTPSTUP             DS    FL1           (PRIVG_USER, PRI_USER)
*                                        indicates normal user,
*                                        system administrator or
*                                        operator
*   indicates normal user, system administrator or operator
EJTPSUTO             EQU   1             TSOS
EJTPSUUS             EQU   2             normal user
EJTPSUOP             EQU   4             operator
```

```
*
EJTPSTF             DS    OXL1
EJTPSTFL            DS    AL1
EJTPSTFI            EQU   X'80'         job held: /HOLD-JOB given
EJTPSTFT            EQU   X'40'         a /CANCEL-JOB command has
*                                       been given
EJTPSTFS            EQU   X'20'         a /CAN-JOB STEPS=*CURR has
*                                       been given
EJTPSTFC            EQU   X'10'         a /CAN-JOB
*                                       STEPS=*ALL-CALENDAR-REPET
*                                       has been given
EJTPSTIE            EQU   X'08'         job is being extracted
EJTPSTTE            EQU   X'04'         special termination for
*                                       jobs-in-extract
EJTPSTFJ            EQU   X'02'         job held: missing cal cat
EJTPSTFR            EQU   X'01'         unused
         ORG   EJTPSTF
EJTPSTOL            DS    AL1
EJTPSTOI            EQU   X'80'         (HOLD) job is held
EJTPSTOR            EQU   X'7F'         unused
         ORG   EJTPSTF+1
EJTPSTRS            DS    XL1           unused
*
*
EJTPCA              DS    OXL32         CALDAT: data of caller
EJTPCACR            DS    CL8           userid that created this task
EJTPCATS            DS    CL4           (ORIG_TSN) TSN of the job
*                                       that created this one
EJTPCAAU            DS    XL16          audit id (chipcard id of
*                                       original dialog job)
EJTPCAJT            DS    X             job type: see JATOTYP in
*                                       macro D$JAT (resp.
*                                       JMS_JO_TYPE_SET in JMSSETI,
*                                       SPL)
EJTPCAFL            DS    AL1
EJTPCAFA            EQU   X'80'         privileged acceptance or not
EJTPCAFC            EQU   X'40'         enter from console or not
EJTPCAFT            EQU   X'20'         rfa servertask or not
EJTPCAFR            EQU   X'1F'         unused
EJTPCARS            DS    XL2           unused
*
*
EJTPED              DS    OXL120        EDAT: batch data
EJTPSPOT            DS    F             time of job acceptance
EJTPEDFN            DS    CL54          (NON_CONV_TASK) name of enter
*                                       (SYSCMD) file
EJTPEDFP            DS    CL4           (ENTER_PSWD) password of
*                                       enter file
```

```
EJTPEDON            DS    CL54          name of the file that was
*                                       copied into an S.IN-file
EJTPEDRC            DS    H             repeat count
EJTPEDFL            DS    AL1
EJTPEDF1            EQU   X'80'         (ENTER_ER) erase enter file
EJTPEDF2            EQU   X'40'         (NOT_ER) do not erase enter
*                                       file
EJTPEDFS            EQU   X'20'         don't count job in job class
EJTPEDFX            EQU   X'10'         S.OUT. files created by
*                                       forked tasks are printed
EJTPJNFE            EQU   X'08'         take JOB_NAME from ENTER-JOB
*                                       command
EJTPNLTM            EQU   X'04'         tasklocal time shall not
*                                       apply
EJTPTDFE            EQU   X'02'         syscmd file is execute only
EJTPSPAR            EQU   X'01'         repeat jobs permit start time
*                                       in the past
EJTPEDRS            DS    XL1           unused
*
*
EJTPPA              DS    OXL204        PARLST: some other data
EJTPPAMN            DS    CL54          (JOB_VAR_NAME) name of
*                                       monitoring JV
EJTPPAMP            DS    CL4           (JV_PSWD) password of
*                                       monitoring JV
EJTPPARP            DS    X             (PRI) run priority for task
*                                       management
EJTPPAJP            DS    X             job priority
EJTPPASL            DS    F             line limit for SYSLST
EJTPPASO            DS    F             card limit for SYSOPT
EJTPPATL            DS    F             (TIME_VALUE) cpu time limit
*                                       for job
EJTPPARS            DS    XL2           unused
EJTPPAJC            DS    CL128         job class parameters
EJTPPAMS            DS    OXL1          message attributes
EJTPPAMD            DS    X             message attributes
         ORG   EJTPPAMS
EJTPPAMB            DS    XL1           unused
         ORG   EJTPPAMS
EJTPPAFM            DS    AL1           Meldungsattribute
EJTPPAM1            EQU   X'C0'         unused
EJTPPAML            EQU   X'20'         copy SYSOUT to SYSLST
EJTPPAM2            EQU   X'10'         unused
EJTPPAMT            EQU   X'08'         terminal hardcopy, unused
EJTPPAMH            EQU   X'04'         display console messsages on
*                                       SYSOUT
EJTPPAMC            EQU   X'02'         Meldungen werden nur in
*                                       Kurzform ausgegeben
```

```
EJTPPAMF             EQU   X'01'         Meldungen werden mit
*                                        erlaeuterndem Text augegeben
         ORG   EJTPPAMS+1
EJTPPAFL             DS    AL1
EJTPPAFF             EQU   X'80'         flush after shutdown
EJTPPAFR             EQU   X'40'         rerun after crash
EJTPPAFC             EQU   X'20'         job is protected against
*                                        involuntary cancel
EJTPPAFI             EQU   X'10'         /HOLD-JOB not allowed
EJTPPAFP             EQU   X'08'         a system logon procedure is
*                                        not executed ($JOBENT only)
EJTPPAFS             EQU   X'04'         no line limit for syslst
EJTPPAFO             EQU   X'02'         no line limit for sysopt
EJTPPAFU             EQU   X'01'         (NTL_REQ) no cpu time limit
*
*
EJTPSR               DS    0XL8          START: various start
*                                        attributes
EJTPSRTI             DS    F             planned start time
EJTPSRTY             DS    X             start type: see JATPJST in
*                                        macro D$JAT (resp.
*                                        JMS_START_SET in JMSSETI,
*                                        SPL)
EJTPSRRS             DS    XL3           unused
*
EJTPMISC             DS    0XL92
*
EJTPCD               DS    0XL92         CALENDAR: data for calendar
*                                        jobs
EJTPCDNA             DS    CL54          name of calendar
EJTPCDSD             DS    CL20          symbolic date in calendar
EJTLIMIT_INDICATOR   DS    X             limit indicator:
*                                        NO=0/COUNTER=1/TIME=2
*                                        NO_LIMIT given or
*                                        LIMIT_COUNT valid or
*                                        LIMIT_TIME valid
EJTPCDRS             DS    XL1           unused
EJTLIMIT             DS    0XL4
EJTPLIMT             DS    F             limit time : JMS time format
         ORG   EJTLIMIT
EJTPLIMC             DS    H             limit count
         ORG   EJTLIMIT+4
*
EJTPLIMS             DS    0XL12         structured limit time
EJTPLIMH             DS    X             hours
EJTPLIMM             DS    X             minutes
EJTPLIMD             DS    CL10          date
*
```

```
*
         ORG   EJTPMISC
*
EJTPSU               DS    OXL60         SUBTASK: temporary data
*                                        during subtask proliferation
EJTPSUOR             DS    CL8           name of subsystem that
*                                        created the subtask
EJTPSUCA             DS    CL7           (CATEG) task start category
EJTPSUSP             DS    X             (PRIVIL) set privileges
*                                        according to
*                                        PRIVILEGE_VECTOR
EJTPSUPV             DS    XL42          (SRPM_DATA) privilege vector
*                                        for subtask
EJTPSUTY             DS    X             (TASK_TYPE) subtask type
EJTPSUCS             DS    X             (CMDCON) is task to be
*                                        connected to SDF?
*
         ORG   EJTPMISC
*
EJTPRB               DS    OXL34         RBP: data for Remote Batch
*                                        Processing
EJTPRBNA             DS    CL8           job name of an RBP job
EJTPRBSS             DS    CL8           (REM_STATION) name of station
*                                        that submitted the RB job
EJTPRBDS             DS    CL8           (DEST_STATION) destination
*                                        station for SPOOL
EJTPRBAU             DS    CL8           (ALT_USER) Who knows?
EJTPRBBS             DS    X             (BTCH_STN) some data for
*                                        spool?
EJTPRBU              DS    OXL1
EJTRB_OUT_DEF        DS    CL1           remote terminal control
         ORG   EJTPRBU
EJTPRBFL             DS    AL1
EJTPRBF1             EQU   X'F8'         unused
EJTPRBFT             EQU   X'04'         send output to remote station
EJTPRBF2             EQU   X'02'         unused
EJTPRBFO             EQU   X'01'         hold spoolout until requested
         ORG   EJTPRBU+1
*
         ORG   EJTPMISC+92
*
EJTPRE               DS    OXL16         REPEAT: repeat job data
EJTPRETY             DS    X             repeat type: see JATPJRT in
*                                        macro D$JAT (resp.
*                                        JMS_REPEAT_SET in JMSSETI,
*                                        SPL)
EJTPRERM             DS    OXL1
EJTPREIN             DS    X             repeat indicator
```

```
         ORG   EJTPRERM
EJTPREIB             DS    AL1
EJTPREMF             EQU   X'FE'          unused
EJTPREMC             EQU   X'01'          monjv is definitively closed
*                                         by jms
         ORG   EJTPRERM+1
EJTPREIV             DS    H              repeat interval
EJTPRETI             DS    F              planned start time of next
*                                         image
EJTPRETL             DS    F              time of last repeat
EJTPRETU             DS    0XL4
EJTPRETS             DS    CL4            tsn of last repeat
         ORG   EJTPRETU
EJTPRETF             DS    A
EJTPRETB             EQU   X'FFFFFFFF'    tsn of last repeat is set
         ORG   EJTPRETU+4
*
*
EJTPJD               DS    0XL34          JDRES: job distribution data
EJTPJDHO             DS    CL8            (SUBMITT_JOB) name of host
*                                         that sent the job
EJTPJDOX             DS    AL1
EJTPJDOE             EQU   X'80'          old parameter EXPRESS
EJTPJDXR             EQU   X'7F'          unused
EJTPJDOM             DS    XL1            old parameter MSG
EJTPJDOP             DS    X              old parameter PRIO
EJTPJDRH             DS    X              repeat hours
EJTPJDRM             DS    X              repeat minutes
EJTPJDSH             DS    X              start hours
EJTPJDSM             DS    X              start minutes
EJTPJDRS             DS    XL1            unused
EJTPJDSD             DS    CL10           start date
EJTPJDDE             DS    XL4            expl default-ind of JAP
EJTPJDDI             DS    XL4            impl default-ind of JAP
*
*
EJTPSQ               DS    0XL8           STATUS QUO: data describing
*                                         rights that are recorded
*                                         during job acceptance and
*                                         used for task initialization
EJTPSQSC             DS    X              (CLASS_OUT)
EJTPSQMC             DS    CL3            (MAX_ALLOWED_CATEGORY)
EJTPSQMP             DS    X              (MAX_PRIORITY) maximum run
*                                         priority
EJTPSQAR             DS    X              (READ) test option
EJTPSQAW             DS    X              (WRITE) test option
EJTPSQOO             DS    XL1            unused
*
```

```
*
EJTPTD                DS    OXL12          TO DELETE: data that should
*                                          be eliminated from the JTBP,
*                                          but are currently used by
*                                          someone
EJTSPOOLIN_TIME_STCK DS     XL8            (SPOOL_IN_TIME_STCK) kann aus
*                                          SPOOLIN_TIME berechnet
*                                          werden
EJTPTDBL              DS    H              (BUFF_PARAM) data available
*                                          from $JCBRW
EJTPTDFL              DS    AL1
EJTPTDFF              EQU   X'80'          unused
EJTPTDFD              EQU   X'40'          syscmd is a /DATA file
EJTPTDFR              EQU   X'3F'          unused
EJTPTDOO              DS    XL1            unused
*
*
EJTPMO                DS    OXL4           MOVE: data for EXTRACT/IMPORT
EJTPMOTS              DS    CL4            TSN of the job before
*                                          EXTRACT/IMPORT
*
EJTEMPTY              DS    XL14           unused
EJTBP#                EQU   *-EJTPIDNT
```

```
         SJMCHKZ MF=D
*************************************************************
* ,SJMCHKZ': DATA FOR THE INTERFACE  ,JOIN-CMD'.            *
*-----------------------------------------------------------*
* THE FOLLOWING DATA IS TRANSFERRED:                        *
*  | ID | DATA  | ID | DATA  | ....  | END_ID |             *
*                                                           *
* THE 1-BYTE IDS AND THE IMPLICIT LENGTHS (WITH THE ID IN   *
* EACH CASE) ARE DEFINED HERE: ID = KZ IN NAMES             *
*************************************************************
*
*************************************************************
* 1. ID FOR THE ,USER PART' OF THE ,JOIN RECORD'.           *
*************************************************************
*                  RECORD KEY
SJKZUSR    EQU   1             ,USERID'
SJKZUSRL   EQU   9
*
*                  GROUP
SJKZGRP    EQU   2             ,GROUP='
SJKZGRPL   EQU   9
*
*                  PASSWORD /PASSWORD OPERATIONS
SJKZPAS    EQU   3             ,PASS='
SJKZPASL   EQU   18            L=(ID,ENCR-ID,PSWORD1,PSWORD2)
SJKZPNO    EQU   0               ENCR-ID: DO NOT ENCRYPT,
SJKZPEN    EQU   1                        ENCRYPT.
*
SJKZECM    EQU   4             -FREE-
*
SJKZPSW    EQU   5             ,PSWORD='
SJKZPWN    EQU   1             TRANSFER VALUES.
SJKZPWM    EQU   2
SJKZPWY    EQU   3
SJKZPSWL   EQU   2
*
*                  SEVER/LOCK INDIKATOR
SJKZSVR    EQU   6             ,LOCK='
SJKZVRY    EQU   1             TRANSFER VALUE
SJKZSVRL   EQU   2
*
*                  PRIMARY CATALOG BLOCK NUMBER
SJKZPBN    EQU   7             ,PBN'
SJKZPBNL   EQU   3
*
*                  DEFAULT CATID
SJKZDPS    EQU   8             ,DEF-PUBSET='
SJKZDPSL   EQU   5
```

```
*
SJKZUPC    EQU   9             ,USER PRIVILEGE CODE'
SJKZADMN   EQU   1             TRANSFER VALUES
SJKZNUSR   EQU   2
SJKZUPCL   EQU   2
*
SJKZTPR    EQU   10            ,TESTPRIV='
SJKZPRY    EQU   1
SJKZPRN    EQU   2
SJKZTPRL   EQU   4             L=(ID,AID1,AID2,Y/N)
*
SJKZCST    EQU   11            ,CSTMP-MAC='
SJKZCSY    EQU   1
SJKZCSN    EQU   2
SJKZCSTL   EQU   2
*
SJKZADT    EQU   12            ,AUDIT='
SJKZADY    EQU   1
SJKZADN    EQU   2
SJKZADTL   EQU   2
*
SJKZENF    EQU   13            ,ENF='       EXTENDED IN V11
SJKZEFY    EQU   1                        DOES NOT MATCH VALUE STORED
SJKZEFN    EQU   2                        IN USER RECORD.
SJKZEFT    EQU   3
SJKZENFL   EQU   2
*
SJKZTPI    EQU   14            ,TPIGNORE='
SJKZTPIL   EQU   2
*
SJKZUSS    EQU   15            ,USER SWITCHES'
SJKZUSSL   EQU   13            L=(KZ,ON-SW,OFF-SW,INV-SW)
*
*                  RESOURCES (USER TEIL)
SJKZADR    EQU   16            ,ADDRSPACE='
SJKZADRL   EQU   3
*
SJKZPUBS   EQU   17            ,PUBSPACE=' OR SPACE-LIMIT
SJKZPUBL   EQU   5
*
SJKZSUS    EQU   19            ,SPACE-USED'
SJKZSUSL   EQU   5
*
SJKZRPG    EQU   20            ,RES-PAGES='
SJKZRPGL   EQU   3
*
SJKZMAR    EQU   21            ,MAX-ACC-REC='
SJKZMARL   EQU   3
```

```
*
*                  MESSAGE HANDLING
SJKZMSG    EQU   22            ,MESSAGE SEARCH='
SJKZTSK    EQU   1
SJKZALL    EQU   2
SJKZMSGL   EQU   2
*
SJKZDML    EQU   23            ,DEF-MSG-LAN='
SJKZDMLL   EQU   2
*
*                  MAILING ADDRESS/ COMMAND FILE NAME
SJKZMAL    EQU   24            ,MAIL='
SJKZMALL   EQU   65
*
SJKZCMF    EQU   25            ,COMMANDS='  NEW: PROFILE ID
SJKZCMFL   EQU   55
*
*****************************************************************
* EXTENSIONS IN V11: QUOTA, TUNING MEASURES                     *
*                       EXTENDED HOST CODE.                     *
*****************************************************************
*
SJKZTSL    EQU   35            ,TEMP-SPACE-LIMIT='
SJKZTSLL   EQU    5
*
SJKZTSU    EQU   36            ,TEMP SPACE USED'
SJKZTSUL   EQU    5
*
SJKZFLI    EQU   37            ,FILE-LIMIT='
SJKZFLIL   EQU    5
*
SJKZFLA    EQU   38            ,FILE AMOUNT'
SJKZFLAL   EQU    5
*
SJKZJVL    EQU   39            ,JV-LIMIT='
SJKZJVLL   EQU    5
*
SJKZJVA    EQU   40            ,JV AMOUNT'
SJKZJVAL   EQU    5
*
SJKZXHC    EQU   41            ,EXTENDED-HOST-CODE='
SJKZXHCL   EQU    9
*
SJKZDTR    EQU   42            ,DMS-TUNING-RESOURCES='
SJKZDTRN   EQU    1               NONE
SJKZDTRP   EQU    2               PAGEABLE
SJKZDTRR   EQU    3               RESIDENT
SJKZDTRL   EQU    2
```

```
*
SJKZPAL    EQU   44            ,'PHYSICAL-ALLOCATION='  :* V12.0
SJKZPAN    EQU   1
SJKZPAA    EQU   2
SJKZPALL   EQU   2
*
SJKZHAS    EQU   45            ,HIGH-AVAILABLE-SPACE=' :* V12.0
SJKZHASL   EQU   5
*
SJKZS1U    EQU   46            ,S1_LEVEL_USED'         :* V12.0
SJKZS1UL   EQU   5
*
SJKZS2U    EQU   47            ,S2_LEVEL_USED'         :* V12.0
SJKZS2UL   EQU   5
*
SJKZHAU    EQU   48            ,'HARDWARE-AUDIT'         :* V13.0
SJKZHAA    EQU   1
SJKZHAN    EQU   2
SJKZHAUL   EQU   2
*
SJKZLAU    EQU   49            ,'LINKAGE-AUDIT'          :* V13.0
SJKZLAA    EQU   1
SJKZLAN    EQU   2
SJKZLAUL   EQU   2
*
*
*****************************************************************
* 2. ID FOR THE ,ACCOUNT PART' OF THE ,JOIN RECORD'.            *
*---------------------------------------------------------------*
* - WHEN MODIFYING AN EXISTING ACCOUNT NUMBER, THE DATA         *
*   ITEMS ARE PASSED INDIVIDUALLY. THIS MODIFICATION DATA       *
*   REFERS TO THE ACCOUNT NUMBER IN ,SJKZACT'!                  *
*****************************************************************
*
*                  ACCOUNT-SPECIFIC PRIVILEGES
SJKZACT    EQU   26            ,ACCOUNT='
SJKZACTL   EQU   9
*
SJKZNTL    EQU   27            ,NTL='
SJKZNTY    EQU   1
SJKZNTN    EQU   2
SJKZNTLL   EQU   2
*
SJKZEXP    EQU   28            ,EXPRESS='
SJKZEXY    EQU   1
SJKZEXN    EQU   2
SJKZEXPL   EQU   2
*
```

```
SJKZINH    EQU   29            ,INHD='
SJKZINY    EQU   1
SJKZINN    EQU   2
SJKZINHL   EQU   2
*
SJKZTTY    EQU   30            ,TTYPL='
SJKZTSTD   EQU   1
SJKZTTP    EQU   2
SJKZTSYS   EQU   3
SJKZTTYL   EQU   2
*
*                  ACCOUNT-SPECIFIC RESOURCES
SJKZTIM    EQU   31            ,TIME='
SJKZTIML   EQU   5
*
SJKZPRI    EQU   32            ,PRIORITY='
SJKZPRIL   EQU   2
*
SJKZCLA    EQU   33            ,CLASS='
SJKZCLAL   EQU   2
*
SJKZPOS    EQU   43            ,POSIX='                :* V11.2 POSIX
SJKZPOY    EQU   1
SJKZPON    EQU   2
SJKZPOSL   EQU   2
*
SJKZDEL    EQU   34            ,DEL='
SJKZDELL   EQU   81            (ALWAYS WHOLE AREA)
*
SJKZMAX    EQU   49            FOR QUERIES, MAXIMUM VALUE
*
SJKZEOF    EQU   255
```

```
         SSVSCB D
         MFPRE MF=D,PREFIX=SCB,MACID=E,ALIGN=D,DNAME=E
SCBE     DSECT ,
               *,##### PREFIX=SCB, MACID=E #####
SCBETOP  DC    Y(SCBELEN)          LENGTH OF SCB
SCBESTDL EQU   X'0520'             STANDARD LENGTH OF THE SCB = 1016
SCBEMAXL EQU   X'0BF8'             MAX SCB LEN (FOR 3.0)   = 3064
         DC    CL2' '              FILLER CHARACTERS
         SSVJTC ,E,SCB
SCBEJTC  DS    0F
*************************************************************************
*                                                                       *
*                         COMMON BLOCK                                  *
*                                                                       *
*************************************************************************
SCBEINDC DC    X'00'               JTBP/SPCB INDICATOR FOR ACCOUNTING
SCBEIJT  EQU   X'00'               JTBP INDICATOR
SCBEISP  EQU   X'01'               SPCB INDICATOR
         DC    XL1'00'             RESERVED FOR FUTURE USAGE
*-----------------------------------------------------------------------*
*                           KEY DEFINITION                              *
*-----------------------------------------------------------------------*
SCBENKEY DS    0CL32               QUEUE FILE KEY FOR EQUISAM FILE
SCBEQUE# DC    X'00'               QUEUE FILE NUMBER
SCBEKEY  DS    0CL5                QUEUE FILE KEY FOR OLD FILE METHOD
SCBEPRI  DC    X'00'               PRIORITY CLASS < LOGON
SCBETSN  DC    F'0'                TASK SEQUENCE # FOR THIS JOB
SCBESEQ# DC    X'00'               QUEUE SEQUENCE#
         DS    CL25                UNUSED
         ORG   SCBENKEY            MESSAGE QUEUE KEY DEFINITION
SCBERQE# DS    X                   QUEUE FILE # FOR MESSAGE QUEUE ONLY
SCBERSTN DS    CL8                 STATION IDENTIFICATION
         DS    CL23                UNUSED
         ORG   SCBENKEY            REPLAY TAPES QUEUE KEY DEFINITION
SCBEQ#   DS    X                   QUEUE FILE # (X'0C')
SCBEQF#  EQU   X'0C'               QUEUE #
SCBESID  DS    CL6                 SESSION-ID FROM "SRD.SESSION-ID"
SCBEVSN  DS    CL6                 VOLUME ON WHICH THE FILE RESIDES
SCBETYPE DS    0XL3                TYPE OF OUTDEV FOR THE FILE
SCBETYP0 DS    X                   1ST BYTE OF THE TYPE
SCBETMM  EQU   X'80'               - MIXED MODE
*
SCBETYP1 DS    X                   2ND BYTE OF THE TYPE
SCBETPU  EQU   X'01'               - PUNCHED REQUESTED
SCBETPRT EQU   X'02'               - PRINTER REQUESTED
*
SCBETYP2 DS    X                   3RD BYTE OF THE TYPE
SCBELP3  EQU   X'01'               - PRINTER 136
```

```
SCBELP6  EQU   X'02'              - PRINTER 160
SCBETND  EQU   X'04'              - ND2
SCBETSD  EQU   X'08'              - SD23
SCBETCD  EQU   X'10'              - CARD PUNCHER
SCBETFD  EQU   X'20'              - FD
*
SCBEFSEQ DS    CL2                FSEQ OF FILE ON TAPE
SCBEIND  DS    X                  INDICATOR BYTE
SCBEICF  EQU   X'01'              - CONTINUATION FILE
*
SCBETPRI DS    X                  PRIORITY FROM ORIGINAL CMD
SCBETTSN DS    CL4                TSN FROM ORIGINAL CMD
SCBESQ#  DS    X                  SEQUENCE # FROM ORIGINAL CMD
         DS    CL7                     UNUSED
SCBEKLEN EQU   32                 KEY LENGTH
*-------------------------------------------------------------------*
*                   LOGON TYPE INFORMATIONS                         *
*-------------------------------------------------------------------*
SCBEJNAM DS    CL8                JOB NAME (RJOB, LOGON OR DEFAULT)
SCBEUSR  DC    XL8'00'            ALPHANUMERIC USERID <LOGON
SCBEACT  DC    XL8'00'            ACCOUNT NUMBER <LOGON
SCBEPSWD DC    XL8'00'            PASSWORD < LOGON
SCBEPVG  DC    X'00'              PRIVILEGED USER
SCBEADM  EQU   X'01'              INDICATES SYSTEM ADMINISTRATOR
SCBEUSER EQU   X'02'              INDICATES NORMAL USER
SCBEOPR  EQU   X'04'              INDICATES OPERATOR
*
SCBESCLA DC    X'00'              SPOOLOUT CLASS DEFINITION
SCBESPCS DS    X                  INITIAL SPACE VALUE OF PRINT CMD
SCBECTCS DS    X                  INITIAL CONTROL CHAR VALUE OF PRINT
SCBECCST DS    X                  CCS TYPE
SCBECCSV DS    X                  CCS VARIANT
*-------------------------------------------------------------------*
*              CATALOG,ACCOUNTING,... INFORMATIONS                  *
*-------------------------------------------------------------------*
SCBEUTID DC    F'0'               USER TID                       002
SCBES#IO DC    F'0'               #I/O DURING SPOOLIN
SCBEUID  DC    2F'0'              /ACCOUNT SUPPLIED INFORMATION
SCBECODE DC    H'00'              PRIMARY CATAL BLOCK NUMBER
SCBEDLOF DC    XL3'00'            DATE LOGOFF WAS PROCESSED
SCBETLOF DC    XL2'00'            TIME LOGOFF WAS PROCESSED
*-------------------------------------------------------------------*
*                        RBP INFORMATIONS                           *
*-------------------------------------------------------------------*
SCBEXBSE DC    X'00'              INDEX OF BATCH ENTRY STATIONS
SCBEXBJE DC    X'00'              INDEX OF BATCH JOB ENTRIES
*-------------------------------------------------------------------*
*                      GENERAL INFORMATIONS                         *
```

```
*-------------------------------------------------------------------------*
SCBEMSG  DC    X'00'              OPTION  BYTE
SCBEFLMG EQU   X'01'               F = FULL MESSAGE OPTION
SCBECDMG EQU   X'02'               C = SHORT MESSAGE OPTION
SCBEHCMG EQU   X'04'               H = HARD COPY OPTION
SCBEHCT  EQU   X'08'               T = TERMINAL'S HARD-COPY
SCBEDSTP EQU   X'10'               --- SWITCH FROM DISC TO TAPE
SCBEMSLO EQU   X'20'               L = LOGGING REQUESTED INDICATOR
SCBEBCPU EQU   X'80'               --- INSUFFICIENT CPU TIME LEFT OR
*                                  --- BIAS TIME VALUE IS SET LOWER
*                                  --- THEN TIME VALUE GIVEN OR
*                                  --- DEFAULTED
SCBESW   DC    X'00'               JTBP SWITCH FIELD
SCBEPVT  EQU   X'01'               SET: ENTER ERASE=YES GIVEN
SCBESCHD EQU   X'02'               SET: JOB IS SCHEDULED
SCBEDVQ  EQU   X'04'               SET: TASK IS ON THE DEVICE QUEUE
SCBENTL  EQU   X'08'               SET: NTL REQUESTED
SCBEENTR EQU   X'10'               JOB CREATED VIA ENTER
SCBERBP  EQU   X'20'               JOB CREATED VIA REMOTE
SCBERCJ  EQU   X'40'               JOB CREATED VIA RCARD
SCBENER  EQU   X'80'               DO NOT ERASE FILE DURING LOGOFF
*                                  ALSO, DO NOT SET UP TCB DURING
*                                  TSN ASSIGNEMENT, ONLY THE JTBP
SCBELRI  DS    X                  MAX PRI ALLOWED FOR AN ACCOUNT #
*
SCBESW2  DC    X'00'              SYSFILE SWITCH FIELD 2
SCBETEMF EQU   X'80'              TEMP FILE USED
SCBETEMJ EQU   X'40'              TEMP JV USED
SCBEEX94 EQU   X'20'              EXIT 94 LOADED
SCBEUFSF EQU   X'08'              UFS FILE
SCBEUFSC EQU   X'04'              UFS COPY CREATED
SCBERSRL EQU   X'02'              RESOURCE LOCATION : ON=SERVER
*                                                     OFF=CLIENT
SCBELIBM EQU   X'01'               LMS LIBRARY MEMBER
*
SCBEPARE DC    X'00'               SPARAM EDIT VALUE
SCBEPARN EQU   X'00'               EDIT= NO
SCBEPARP EQU   X'01'               EDIT= PARTIAL
SCBEPART EQU   X'02'               EDIT= TOTAL
*
SCBEMLN  DC    CL1' '              MESSAGE LANGUAGE
SCBEMLNS DC    CL1' '              MESSAGE LANGUAGE STANDARD
SCBECNAM DS    CL8                 JOBCLASS NAME
SCBEJPRI DS    X                   JOB PRIORITY
SCBEHLDI DS    X                   JOB HOLD INDICATOR
SCBEHLDY EQU   X'80'                - JOB IN HOLD
SCBECENT DS    X                   CENTURY FOR THE DATE            036
SCBESW3  DS    X                   INDICATOR 3                     042
```

```
SCBEFTIP EQU   X'01'                - FT IN PROGRESS(NO DELETE AT
SCBESICK EQU   X'02'                - SINIX CLIENT KEEP              061
SCBEFILT EQU   X'04'                - JOB FILTERED                   081
SCBEUSTT EQU   X'08'                - USER TRANSLATION TABLE         081
SCBESTPR EQU   X'10'                - START-PROC NE IMMEDIATE        082
SCBERDCM EQU   X'20'                - RDIR COMMAND FROM SYSFILE      082
*                                     CLIENT STARTUP
SCBEFPP# DS    F                   TOTAL PPAGE# IN A FAMILY          005
SCBECLEN EQU   *-SCBEJTC
SCBETLP  EQU   X'40'            - 3337 TYPE (SEE &P.TYP2)
SCBETHP  EQU   X'80'            - HP TYPE (SEE &P.TYP2)
         EJECT
         SSVSCBE ,E
         MFPRE MF=,PREFIX=SCB,MACID=E,ALIGN=F,DNAME=ESCB             010
SCBESCB  DS    0F
*********************************************************************
*                         SPOOL SUBSYSTEM BLOCK                     *
*********************************************************************
*
*
*******************************
*     SOURCE INFORMATION      *
*******************************
*
*
*-------------------------------------------------------------------*
*                LOCAL AND REMOTE SPOOL COMMON FIELDS               *
*-------------------------------------------------------------------*
SCBESCB# DS    CL3                  SCB VERSION NUMBER
SCBEV30A EQU   X'100'           START SCB VERSION NUMBER OF VERSION 3.0A
SCBESCBV EQU   X'103'           CURRENT SCB VERSION NUMBER, 95-06-07
SCBELST# DS    X                    SYSLST NUMBER
SCBEOTSN DC    F'0'             TSN OF TASK REQUESTOR
SCBEFILN DC    A(0)             INTERNAL FILENAME OF FILE TO BE PRINTED
SCBESIZE DS    F                FILE SIZE (# OF PAM PAGES)           100
SCBERPSW DC    XL4'00'          FILE READ PASSWORD
SCBEFID  DC    CL54' '          FILE TO BE PRINTED(OR FILE# FOR EAM)
SCBEFTYP DC    X'00'            FILE TYPE TO BE OUTPUTTED
SCBESEAM EQU   X'01'            SYSTEM FILE(EAM)
SCBEOMF  EQU   X'02'            OBJECT MODULE FILE(EAM)
SCBESOUT EQU   X'04'            SYSOUT
SCBESLST EQU   X'08'            SYSLST
SCBETEMP EQU   X'10'            TEMPORARY FILE                       SBC
SCBESOPT EQU   X'20'            SYSOPT                               SBC
SCBEUEAM EQU   X'40'            USER CREATED EAM FILE
SCBEFAM  EQU   X'80'            FIRST MEMBER OF A FAMILY
*
SCBEFFCB DC    X'00'            FILE TYPE
```

```
SCBEFSAM EQU   X'00'         SAM FILE
SCBEFTAP EQU   X'01'         FILE RESIDES ON TAPE                    101
SCBEFEAM EQU   X'20'         EAM FILE (TO KEEP FOR COMPATIBILITY)
SCBEFISA EQU   X'40'         ISAM FILE
SCBEFBTA EQU   X'80'         BTAM FILE
SCBEFPAM EQU   X'C0'         PAM FILE
*
SCBERTYP DC    X'00'         FILE RECORD FORMAT FROM CATAL
SCBERVAR EQU   X'02'         VARIABLE LENGTH RECORDS(V-TYPE)
SCBERFIX EQU   X'04'         FIXED LENGTH RECORDS(F-TYPE)
SCBERUND EQU   X'06'         UNDEFINED RECORDS
*
SCBEBITF DS    X                    BIT MAP INDICATOR
SCBELOER EQU   X'80'                LOCK ERROR
SCBECKAV EQU   X'40'                CHECKPOINT APA AVAILABLE
SCBEAPPL EQU   X'20'                ON : APPLICATION NOT ALLOWED     011
SCBEBITN EQU   X'01'                DO NOT RESET BIT MAP
*
SCBERSIZ DS    H                   RECSIZE
SCBEBSIZ DS    H                   BLOCKSIZE
*
SCBEMAIL DS    CL64          MAILING ADDRESS
*
SCBEFOBS DS    H             FOB SIZE.
*
         DS    0F                                                    102
         DS    H             EAM TAPE FILENAME (TO KEEP FOR COMPAT.)
SCBEOFIL DS    X                  ORIGINAL FILE SPECIFICATIONS
SCBER164 EQU   X'80'              RECORD 164 PRESENT FOR LIB.ELEM.
SCBEPEF1 EQU   X'40'              FCB TYPE OF PLAM ELEMENT
SCBEPEF2 EQU   X'20'                   IDEM
SCBEASAC EQU   X'10'              ASA-CODE SPECIFIED FOR FILE
SCBEPGMO EQU   X'08'              PAGE-MODE
SCBEIAS6 EQU   X'04'         $SYSSPOOL IMAGE APA FILE (600 DPI)     022
SCBEIAU6 EQU   X'02'         $USER IMAGE APA FILE (600 DPI)         022
SCBEOEAM EQU   X'01'              EAM FILE
*
SCBEIMFL DS    X             IMAGE ND OR HP FILE USED
SCBEIMNO EQU   X'00'         NO INFO (OLD SCB)
SCBEIMUS EQU   X'01'         $USER IMAGE HP FILE
SCBEIMTS EQU   X'02'         $TSOS IMAGE HP FILE
SCBEIDUS EQU   X'04'         $USER IMAGE ND FILE
SCBEIDTS EQU   X'08'         $TSOS IMAGE ND FILE
SCBEIAUS EQU   X'10'         $USER IMAGE APA FILE
SCBEIASY EQU   X'20'         $SYSSPOOL IMAGE APA FILE
SCBEIAU3 EQU   X'40'         $USER IMAGE APA FILE (300 DPI)         022
SCBEIAS3 EQU   X'80'         $SYSSPOOL IMAGE APA FILE (300 DPI)     022
*
```

```
SCBE#VSN DS    H           # OF VSN'S IN VSNLIST (FIELD SCBETPVL) 102
SCBEFSQ  DS    H           FSEQ FOR TAPE PROCESSING               102
SCBERTPD DS    H           RETENTION PERIOD                       102
SCBESTSN DS    CL4         TSN REPLAY SAVE AREA                   102
SCBETPTY DS    X           TAPE TYPE                              102
SCBETPR  EQU   X'01'       * REPLAY TAPE                          102
SCBETPUS EQU   X'02'       * USER TAPE (DEVIN=STAPE)              102
SCBETPD  EQU   X'04'       * DMS TAPE (DEVIN=TAPE)                102
SCBETPP  EQU   X'08'       * POOLER TAPE                          102
*
SCBETFL  DS    X           TAPE FLAG                              102
SCBETFLC EQU   X'01'       * TAPE PROCESSED IN COPY MODE          102
SCBETFLD EQU   X'02'       * TAPE PROCESSED IN DIRECT MODE        102
SCBETFLI EQU   X'04'       * INPUT TAPE                           102
SCBETFLO EQU   X'08'       * OUTPUT TAPE                          102
SCBETFLF EQU   X'10'       STATE=FOREIGN
SCBETFLM EQU   X'80'       MIXED MODE
*
SCBETD   DS    X           TAPE DENSITY                           102
SCBETY   EQU   X'B0'       * TAPE = YES
SCBET9N  EQU   X'B1'       * TAPE = T9N
SCBET9P  EQU   X'B2'       * TAPE = T9P
SCBET9G  EQU   X'B4'       * TAPE = T9G
*
SCBEPBLK DS    X           BLOCKSIZE FOR POOLER TAPE              102
SCBETPVL DS    4CL6             VSN LIST FOR CURRENT FILE
SCBETPLG EQU   *-SCBETPVL       VSN LIST LENGTH
SCBECHD1 DS    X           CHARS DENSITY FOR ND2 OFF LINE
SCBELPI  EQU   X'C0'       LINES PER INCH
SCBECPI  EQU   X'30'       CHARS PER INCH
SCBE12L  EQU   X'80'       12 LPI
SCBE8L   EQU   X'40'       8 LPI
SCBE6L   EQU   X'00'       6 LPI
SCBE15C  EQU   X'30'       15 CPI
SCBE12C  EQU   X'20'       12 CPI
SCBE10C  EQU   X'10'       10 CPI
*
SCBECHD2 DS    X           CHARS2 DENSITY
SCBECHD3 DS    X           CHARS3 DENSITY
SCBECHD4 DS    X           CHARS4 DENSITY
SCBEFOBN DS    CL4         FOB/FOBPOOL NAME
SCBEUPDI DS    X           DPRINT INDICATOR
SCBEFDIS EQU   X'80'       - FILE CAN BE DISTRIBUTED
SCBEPVCS EQU   X'40'       - PVS SHARED BETWEEN CLIENT AND SERVER
SCBEPVCG EQU   X'20'       - PVS SHARED BETWEEN CLIENT AND GATEWAY
SCBEBLKT EQU   X'10'       - BLOCKTYPE = STD
SCBEERAU EQU   X'08'       - ERASE OF FILE AUTHORIZED
SCBESTUP EQU   X'04'       - STARTUP
```

```
SCBELCK  EQU   X'02'         - LOCK
SCBEDEL  EQU   X'01'         - DELETE
*
SCBEFTST DS    X             FILE TRANSFER STATUS BYTE
SCBEFTNK EQU   X'01'         - FT ENDED NOT OK
SCBEFTOK EQU   X'02'         - FT ENDED OK
SCBEFTIN EQU   X'04'         - FT IN PROGRESS
SCBEFTRQ EQU   X'08'         - FT REQUESTED
SCBEFTWT EQU   X'10'         - FT WAITING
SCBEFTRE EQU   X'20'         - FT RECOVERY                          074
*
SCBEJBTY DS    X             JOB TYPE
SCBEJBSP EQU   X'80'         - SPOOL JOB (LOCAL)
SCBEJBDP EQU   X'04'         - DPRINT JOB (LOCAL)
SCBECPCL EQU   X'02'         - JOB SENT TO ANOTHER HOST (CLIENT COPY)
SCBECPSV EQU   X'01'         - JOB COMING FROM ANOTHER HOST (SERVER C.)
*
SCBECLTT DS    X             CLUSTER TYPE
SCBECLHO EQU   X'00'         - LOCAL CLUSTER
SCBECLSX EQU   X'01'         - SINIX CLUSTER
SCBECLBS EQU   X'02'         - BS2000 CLUSTER
*
***********************************
*     DESTINATION INFORMATION     *
***********************************
*
*
*-------------------------------------------------------------------*
*                  LOCAL SPOOL SPECIFIC FIELDS                      *
*-------------------------------------------------------------------*
SCBEODEV DC    H'0'          OUTPUT DEVICE MNEMONIC
SCBEDENT DC    X'00'         OUTPUT DEVICE TYPE
SCBEPRT  EQU   X'20'         PRINTER
SCBEPUN  EQU   X'30'         PUNCH
SCBEPFD  EQU   X'90'         FLOPPY DISK
SCBETAP  EQU   X'B0'         TAPE
SCBEDESF DS    X             DESTINATION FLAG
SCBEDGRO EQU   X'01'         DEVICE GROUP
SCBERDIR EQU   X'02'         COMING FROM RDIR
SCBEDIST EQU   X'04'         JOB SCHEDULED ON A REMOTE SERVER (DP)
SCBEIDOM EQU   X'08'         JOB IDOM                               078
SCBEBSDC EQU   X'10'         SBSD CALLER                            078
SCBELHST EQU   X'20'         LONG HOSTNAME SUPPORT                  078
*-------------------------------------------------------------------*
*               REMOTE SPOOL SPECIFIC FIELDS                        *
*-------------------------------------------------------------------*
SCBERUID DC    CL8'00'       REMOTE OUTPUT USERID                   211
         ORG   SCBERUID
```

```
SCBENUID DS    CL8          NEW REMOTE OUTPUT USERID
************************************
*     REQUIREMENT  INFORMATION      *
************************************
*
*-------------------------------------------------------------------------*
*      PRINT COMMAND PARAMETERS-COMMON TO LOCAL AND REMOTE                *
*      (AND ALSO COMMON WITH PUNCH COMMAND FIELDS)                        *
*-------------------------------------------------------------------------*
SCBEPNAM DS    XL8           PRINT OR PUNCH JOB NAME
SCBECNTR DS    X             RECORD BYTE FOR CONTROL PHYSICAL
SCBEREAD DS    X             REMAIN ADVANCE FROM PCC
SCBESTRT DC    F'0'          STARTNO:BINARY LHE RECORD POSITION
SCBESALL EQU   X'00'         ZERO=BEGIN AT FIRST POSITION/BLOCK
SCBEEND  DC    F'0'          ENDNO:BINARY RHE STOP RECORD POSITION
SCBEEALL EQU   X'00'         ZERO=END AT 132 OR 160 FOR PRINTER
SCBEFRM  DC    XL6'40'       FORM SIZE CODE
SCBEFRMS EQU   C'STD'        INDICATE SYSTEM STANDART OR DEFAULT
SCBEHREC DC    H'0'          FREQUENCY OF HISTORICAL RECORDS
SCBEHPRT EQU   336           PRINTER DEFAULT IS 336 LINES
SCBEHPUN EQU   26            PUNCH DEFAULT IS 26 CARDS
*
SCBEER   DC    X'00'         ERASE OPTION
SCBEERN  EQU   X'00'         DO NOT ERASE FILE AFTER PROCESSING
SCBEERD  EQU   X'FE'         DESTROY OPTION
SCBEERY  EQU   X'FF'         ERASE FILE AFTER PROCESSING
*
SCBEERR  DC    X'00'         READ ERROR OPTION
SCBEERT  EQU   X'00'         TERMINATE UPON READ ERROR
SCBEERS  EQU   X'40'         SKIP BAD BLOCK UPON READ ERROR
SCBEERA  EQU   X'80'         OUTPUT BAD BLOCK AS IT IS
SCBEERF  EQU   X'C0'         OUTPUT BAD BLOCK AND FLAG IT
*-------------------------------------------------------------------------*
*     PRINT COMMAND PARAMETERS-COMMON TO LOCAL AND REMOTE                 *
*     (BUT NO LONGER COMMON WITH PUNCH COMMAND FIELDS)                    *
*-------------------------------------------------------------------------*
SCBELINE DC    H'0'          NUMBER OF LINES MAXIMUM PER PAGE       210
SCBEPSIZ DC    Y(132)        PRINTER SIZE IN BYTES PER LINES
SCBEP132 EQU   132           INDICATES 132 CHARACTER PRINTER
SCBEP160 EQU   160           INDICATES 160 CHARACTER PRINTER
*
SCBEHDR  DC    X'00'         HEADER INFORMATION ON EACH PAGE
SCBEHRDN EQU   X'00'         NOTHING IS PRINTED AS PART OF HEADER
SCBEHDRP EQU   X'04'         PAGE NUMBER PART OF HEADER
SCBEHDRS EQU   X'10'         STANDARD HEADER-ID,PG,TSN,DAT,FNAME
SCBEHDRD EQU   X'40'         DATE PART OF HEADER
SCBEHDRT EQU   X'80'         FIRST LOGICAL RCD=HEADER/PAGE
*
```

```
SCBEHEAD DC    X'00'         CMD INFO                                  212
SCBEHNCS EQU   X'00'                                                   212
SCBELSPE EQU   X'01'         LINE SPECIFIED IN PRINT REQUEST           210
SCBEHNSP EQU   X'02'         JOB NOT REALLY SUSPENDED (FROM)           LPM
SCBESHSP EQU   X'04'         SHIFT SPECIFIED IN PRINT REQUEST          212
*
SCBESPRT DC    X'00'         SEPARATOR PAGE TO BE OUTPUT
SCBESYES EQU   X'00'         YES
SCBESNO  EQU   X'01'         NO
*
SCBETRLR DC    X'00'         TRAILER PAGE NECESSARY
SCBETYES EQU   X'00'         YES
SCBETNO  EQU   X'01'         NO
*
SCBEEDIT DC    X'01'         EDIT CHARACTER OR SPACING
SCBEEDTE EQU   X'00'         SPACE=E(1ST RCD CHAR=PRNT CNTRL)
SCBEEDT1 EQU   X'01'         LINES=1(SPACE 1 LINE BETWEEN RCDS)
SCBEEDT2 EQU   X'02'         LINES=2(      2                  )
SCBEEDT3 EQU   X'03'         LINES=3(      3                  )
SCBEEDTN EQU   X'04'         SPACE=NO (FOR RSO 2.0)
*
SCBEETYP DC    X'00'         CONTROL CHAR TYPE
SCBEEIBM EQU   X'01'         CONTROL CHAR IS IBM
SCBEEASA EQU   X'02'         CONTROL CHAR IS ASA
SCBEPALN EQU   X'04'         PARAM LINE SPECIFIED
SCBEETE  EQU   X'80'         SPACE=TE (FOR RSO 2.0)
SCBEETI  EQU   X'81'         SPACE=TI       "
SCBEETA  EQU   X'82'         SPACE=TA       "
*
SCBECSW1 DS    XL1                    OUTPUT CONTROLLER SWITCH 1   213
SCBENOER EQU   X'04'                  S: NO ERASE ALLOWED          213
SCBENODL EQU   X'02'                  S: NO DEL. FROM SHUT. Q      213
*
SCBECSW2 DS    XL1                    OUTPUT CONTROLLER SWITCH 2   213
SCBEIOER EQU   X'01'               S: I/O ERROR HAPPENED ON PRINTER    213
SCBEDMSE EQU   X'04'               S: DMS ERROR HAPPENED ON INPUT FILE 213
SCBEUSRE EQU   X'08'               S: USER ERROR HAPPENED              213
SCBERTDR EQU   X'80'               S: REPLAY TAPE DIRECT               213
*
SCBEEXIT DC    X'00'                  EXIT CALL REQUEST            210
SCBEEX95 EQU   X'02'                  OUTPUT RECORD                210
SCBEEX90 EQU   X'01'                  INPUT RECORD                 210
SCBEBIN  DC    X'00'         BINARY OPTION
SCBEXPBN EQU   X'00'         GRAPHIC ONLY IS OUTPUTTED
SCBEXPBY EQU   X'02'         BOTH HEX AND GRAPHIC ARE OUTPUTTED
*
SCBEPPFL DC    F'0'          NO OF PAM PAGES IN FILE
SCBEPPPR DC    F'0'          NO OF PAM PAGES PRINTED
```

```
SCBEFRNB DC    F'0'          VALUE OF FROM OR LAST PARAM
SCBETONB DC    F'0'          VALUE OF TO PARAM
SCBESEL  DC    X'00'         SELECTIVE PRINT OPTION FEATURE INDICATOR
SCBENSEL EQU   X'00'         NO SELECTIVE PRINT FEATURE
SCBESELA EQU   X'02'         LAST OPTION
SCBESELT EQU   X'04'         TO OPTION
SCBESELF EQU   X'08'         FROM OPTION
SCBESELL EQU   X'40'         TYPE=L
SCBESELP EQU   X'80'         TYPE=P
*
SCBEFCPY DC    X'00'         NUMBER OF FILE COPIES
SCBECOPY EQU   SCBEFCPY
SCBEPCPY DC    X'00'         NUMBER OF PAGE COPIES
*--------------------------------------------------------------------*
*        PRINT COMMAND PARAMETERS-ONLY MEANINGFULL FOR LOCAL          *
*--------------------------------------------------------------------*
SCBES2   DC    X'00'         SWITCH 2
SCBESD   EQU   X'01'         JOB MAY BE DIRECTED ON SD23
SCBEND   EQU   X'02'         JOB MAY BE DIRECTED ON ND2
SCBEMEX  EQU   X'04'         MXM EXTENSION INDIRECTLY REQUESTED
SCBECVFB EQU   X'08'         CHECK VFB INDIRECTLY REQUESTED
SCBEMND  EQU   X'10'         JOB MUST BE DIRECTED ON ND2
SCBENOLD EQU   X'20'         JOB MAY NOT GO ON OLD PRINTER
SCBE160  EQU   X'40'         JOB MUST BE REDIRECTED ON 160        CLEAN
SCBE37   EQU   X'80'         MAY ON 3337
*
SCBELOOP DS    XL3           LOOP PARAMETER                         102
SCBEFT1  DS    XL3           FONT 1 KEY                             102
SCBEFT2  DS    XL3           FONT 2 KEY                             102
SCBEFT3  DS    XL3           FONT 3 KEY                             102
SCBEFT4  DS    XL3           FONT 4 KEY                             102
SCBEDIA  DC    XL2'00'       DIA IDENTIFICATION
SCBEIMAG DC    XL4'00'       IMAGE IDENTIFICATION-ND2 PREFIX
SCBESHIF DC    X'00'         SHIFT COLUMN DELAY
SCBECC   DC    X'00'         CONTROL PARAM
SCBECCN  EQU   X'01'         CONTROL = NO
SCBECCP  EQU   X'02'                 = PHYSICAL
SCBEC12  EQU   X'08'                 = 12LPI
SCBECCL  EQU   X'04'                 = LOGICAL
SCBELNMD EQU   X'10'                 = LINEMODE                     210
SCBECCTR EQU   X'20'                 = TRANSPARENT
SCBECAPA EQU   X'40'                 = APA
*--------------------------------------------------------------------*
*            SQ COMMAND AND ABNORMAL I/O MSG PARAMETERS               *
*--------------------------------------------------------------------*
SCBEAND  DC    X'00'         ACTION INDICATOR                       102
SCBEANDR EQU   X'01'         RESPOOL SPECIFIED                      102
SCBEANDS EQU   X'02'         SUSPEND SPECIFIED                      102
```

```
SCBEANDH EQU   X'04'         HOLD "H" SPECIFIED                    102
SCBEANDK EQU   X'08'         HOLD "K" SPECIFIED                    102
SCBEIMEX EQU   X'10'         MPVS IMPORT-EXPORT                   MPVS
*
SCBESUSI DC    X'00'         SUSPEND INDICATOR
SCBESUSL EQU   X'01'         < SPECIFIED
SCBESUSP EQU   X'02'         P SPECIFIED
SCBESUSB EQU   X'04'         B SPECIFIED
SCBESUSS EQU   X'08'         S SPECIFIED
SCBESUSM EQU   X'10'         PAGE # SPECIFIED
SCBESUSY EQU   X'20'         PRIORITY SPECIFIED                    102
SCBECRLG EQU   X'40'         - DEFAULT FORM FROM GEN REC           070
*
SCBELPHC DS    CL3           REQUESTED CROSS LOOP FOR HP (ROTAT.)
SCBESUSG DC    F'0'          PAGE # SPECIFIED IN THE CMD
SCBESUSV DC    X'00'         PRIORITY SPECIFIED IN THE CMD         102
SCBESEC# DS    CL2                SECOND # AT PRINT START
*-------------------------------------------------------------------*
*             SPECIAL PARAMETERS FOR PUNCH FLOPPY DISC               *
*-------------------------------------------------------------------*
SCBEFDNM DS    CL8           FILENAME SPECIFIED
SCBECH#P DS    X             ORDER# OF CHARS REQUESTED IN POOL (HP)
SCBEFDSC DC    H'0'          NUMBER OF RESERVED SECTORS
SCBEFDRP DC    H'0'          RETENTION PERIOD
SCBERCZ  DS    H                  RECSIZE FROM PUNCH CMD
SCBEFDCB DC    X'00'         CONTROL BYTE
SCBEFDBP EQU   X'01'         BYPASS=Y
SCBEFDAC EQU   X'02'         ACCESS=N
SCBEFDWP EQU   X'04'         WRITEPR=Y
SCBEFDSU EQU   X'08'         SKEL=US
SCBEFDSN EQU   X'10'         SKEL=NS
SCBEFLST EQU   X'20'         REPORT REQUESTED                     CLEAN
*
SCBEFDB2 DC    X'00'         CONTROL BYTE 2                        102
SCBEFDNO EQU   X'01'         OWNERID = NO                          102
*                                                                  103
SCBEFDB3 DC    X'00'         CONTROL BYTE 3                        103
SCBEFDSA EQU   X'01'         JOB COPIED SHUTDOWN->ACTIVE Q         103
*                                                                  103
SCBESAER DS    CL1           SAVE ERASE OPTION
*-------------------------------------------------------------------*
*                 VSN LIST USED BY FLOPPY DISCS                      *
*-------------------------------------------------------------------*
SCBEFDVS DS    10CL6          VSN LIST FOR FLOPPY DISC
SCBEFDLG EQU   *-SCBEFDVS     VSN LIST LENGTH
```

```
*
*************************************
*     CONTROL-STATUS INFORMATION      *
*************************************
*
*-------------------------------------------------------------------------*
*                        COMMON INFORMATIONS                              *
*-------------------------------------------------------------------------*
SCBELHI  DS    X             LOOP LENGTH IN 1/6 INCH
SCBELHIC DS    X             CROSS LOOP LENGTH IN 1/6 INCH
SCBEROE  DS    A             @ OF ROE                                  102
SCBECOUN DC    F'0'          SAVED LINES/CARDS COUNT(AFTER /SQ)
SCBEIO   DC    F'0'          SAVED I/O COUNT ON PUB DEVICE(AFTER /SQ)
SCBECNT  DC    F'0'          LINES/CARDS # TO PRINT/PUNCH
SCBEDSPO DC    XL3'00'       DATE JOB SPOOLED OUT
*                            UNUSED (SCBETSPO) FOR COMPATIBIL.
SCBECTTY DS    X                CONTROL TYPE
SCBECTCP EQU   X'01'            = COMPATIBLE
SCBECTHP EQU   X'02'            = HP
SCBECEDA DS    X             CENTURY FOR DATE JOB SPOOLED OUT END
SCBEDSPF DC    XL3'00'       DATE JOB SPOOLED OUT FINISH
*                               UNUSED (SCBETSPF) FOR COMPATIBIL.  211
SCBEPTYP DS    0XL2          PRINTER TYPE
SCBEPTY1 DS    X                FIRST BYTE
SCBEPTY2 DS    X                SECOND BYTE
SCBEPT65 EQU   X'08'            = LP65
SCBEPTHP EQU   X'04'            = HP
SCBEPTAP EQU   X'02'            = APA
SCBEPTAN EQU   X'01'            = ANY
*
SCBEEPCD DS    X             ERROR PAGE CODE                           102
SCBEDMSC EQU   X'01'         DMS ERROR CODE
SCBERTPR EQU   X'02'         ERROR PAGE INDICATOR
SCBESPND EQU   X'04'         SUSPENDED
SCBEHLD  EQU   X'08'         HOLDED
SCBECNCL EQU   X'10'         CANCELLED
SCBEWCB  EQU   X'20'         WRONG CONTROL BYTE
SCBELTR  EQU   X'40'              LINE TRUNCATION OCCURED
SCBETMSG EQU   X'80'         SCBEEPMS IS A PRINTABLE MSG
*
SCBEEPMS DS    CL40          ERROR PAGE MESSAGE                        102
SCBEDSCD DS    CL1           DELAY SCHEDULING BYTE
SCBENSCD EQU   X'FF'         DELAY SCHEDULING IN EFFECT
SCBEFMND DS    X             FAMILY INDICATOR
SCBEFFM  EQU   X'01'         -FIRST MEMBER OF FAMILY
SCBEFMNQ EQU   X'02'         -ALL FAMILY NOT YET IN Q
SCBEFBIT EQU   X'04'         -DO NOT RESET BIT MAP
```

```
*-------------------------------------------------------------------*
*                  LOCAL SPOOL SPECIFIC FIELDS                      *
*-------------------------------------------------------------------*
SCBEROTV DS    X             ROTATION VALUES
SCBER00  EQU   X'F0'         0   DEGREE
SCBER90  EQU   X'F1'         90  DEGREES
SCBER180 EQU   X'F2'         180 DEGREES
SCBER270 EQU   X'F3'         270 DEGREES
SCBER018 EQU   X'F4'         (0,180) DEGREES
SCBER927 EQU   X'F5'         (90,270) DEGREES
SCBE1800 EQU   X'F6'         (180,0) DEGREES
SCBER279 EQU   X'F7'         (270,90) DEGREES
SCBERYES EQU   C'Y'          ROTATION=YES
SCBERNO  EQU   X'00'         ROTATION=NO
*
SCBESAMP DC    H'0'          SAMPLE=VALUE USED FOR FORMS ALIGN.
SCBENSAM EQU   X'00'         ZERO=NO FORMS ALIGNMENT
SCBEXPND DC    X'00'         SQ/CANCEL INDICATOR
SCBEXINT EQU   X'08'         OTHER RESPOOL/SUSPEND NOT ALLOWED
SCBEXPC  EQU   X'10'         CANCEL SPOOL-OUT W/NO SHUTDOWN ENTRY
SCBEXPCH EQU   X'20'             SPOOL SHUTDOWN IN PROGRESS         210
*
SCBECPL  DC    X'00'         SPOOLOUT COMPLETION SWITCH
SCBELOK  EQU   X'00'         NORMAL COMPLETION
SCBELCAN EQU   X'01'         SPOOLOUT TASK CANCELLED
SCBELER1 EQU   X'02'         ABNORMAL OUTPUT I/O TERMINATED TASK
SCBELER2 EQU   X'04'         ABNORMAL INPUT I/O TERMINATED TASK
SCBELRES EQU   X'08'         TASK WAS STOPPED AND RESPOOLED
*
SCBEDIRE DS    H             DISPLACEMENT IN RECORD (MLR)
SCBEHRC  DS    0F            SKIP TYPE HISTORICAL RECORD
SCBEDISP DC    H'0'          DISPLACEMENT WITHIN I/P RECORD
SCBEPOS  DC    H'0'          # OF GET LOOPS ON I/P RECORD
SCBESELC DC    F'0'          SEL PRINT CNT(#LINES)PAGES CORRESPONDING
*                            TO THE I/O COUNT FOR THE SUSPEND FEATURE)
SCBESAVP DC    F'0'          SAVED CURRENT PAGE# FOR HEADER(SQ)
SCBETFLN DC    XL12'00'      NEW FILE NAME GIVEN TO THE FILE WHEN CATA-
*                            LOGING FROM TAPE TO DISC
*-------------------------------------------------------------------*
*                       ND2 SPECIFIC FIELDS                         *
*-------------------------------------------------------------------*
SCBEREC  DC    F'0'          RECORD # (HIST. RCD)
SCBEUF   DC    F'0'          BUFFER # (HIST. RCD)
SCBELNA  DC    F'0'          LINE ADDR (HIST. RCD)
SCBEVFBL DC    X'00'         # OF LINES FROM THE VFB
SCBECH1  DS    X                  FONT# & DENSITY OF 1ST SET 12LPI
SCBECH2  DS    X                  FONT# & DENSITY OF 2ND SET 12LPI
SCBEDEFV DC    X'00'         DEFAULT VALUES GIVEN BY THE SPOOL
```

```
SCBECHAR EQU   X'80'         CHARS
SCBELPRO EQU   X'40'         LOOP
SCBECLPD EQU   X'20'         CROSS LOOP
*
SCBEFNAM DS    CL10          REPLAY FILE MNEMONIC
SCBEIDEV DS    H             INPUT DEVICE MNEMONIC
SCBEVBSD DC    X'00'         # OF LINES FROM VFB FOR SD23
SCBEVBLP DC    X'00'         # OF LINES FROM VFB FOR 3337
SCBEVBND DC    X'00'         # OF LINES FROM VFB FOR ND2
SCBELCH1 DC    X'00'         NUMBER OF LINES BEFORE CHANEL 1
SCBEITN  DC    F'0'          ITN (USABLE BY STATUS ONLY ! )
*-------------------------------------------------------------------*
*                REMOTE CONTROL-TO BE UPDATED IN 1.2                *
*-------------------------------------------------------------------*
SCBEPAGE DC    F'0'          CURRENT PAGE# FOR HEADER
SCBECHPN DS    CL4           CHARS POOL NAME
SCBECRLO DS    CL3           CROSS LOOP PARAMETER
SCBEFDNA DS    C             FD NAME FROM DEVICE PARAMETER
SCBE3171 EQU   X'00'         DEVICE=FD3171 (0 FOR COMPATIBILITY)
SCBEFACO EQU   X'9B'         FD7540/7 (FALCON) REQUESTED
SCBEHPID DS    CL4           USED HPFILE CODED FILE ID.
SCBECHAN DS    H             CHARS NUMBER IN POOL
SCBESTID DS    X             STREAM INTERNAL ID (SLOT SCHEDULING)
SCBELOID EQU   X'00'         STREAM FOR LOCAL SCHEDULING
SCBERSID EQU   X'01'         STREAM FOR RSO SCHEDULING
SCBESPRI DS    X                  PRIORITY BEFORE SCHEDULING
*-------------------------------------------------------------------*
*                       TEMPORARY FOR TAPES                         *
*-------------------------------------------------------------------*
SCBEFLTP DS    X             FLAG BYTE FOR TAPES ****              102
SCBEUSTP EQU   X'11'         * USER TAPE
SCBERLTP EQU   X'10'         * REPLAY TAPE
SCBERRCD DS    CL8           RSO PRINTER ERROR CODE
SCBEDFES DS    X             TO SAVE DFER (RBP CONTROL SWITCH).
*
SCBEDSTN DS    CL8           DESTINATION STATION            JSI &  211
SCBEVBHC DS    H             # OF LINES FROM CROSS VFB FOR HP
*-------------------------------------------------------------------*
*                     REMOTE CONTROL SPOOL V2                       *
*-------------------------------------------------------------------*
SCBEINF  DS    0F            HISTORICAL RECORD                     211
SCBEDMEC DS    CL8               DCAM ERROR CODE (..SPOL:..DMEC)   211
SCBERJBN DS    CL8                     REMOTE JOB NAME             211
SCBERINF DS    X                 REMOTE INFORMATION                211
SCBETERM EQU   X'01'     SET: JOB ABNORMALLY TERMINATED             211
SCBERPRT EQU   X'02'     SET: PRINT TO THE REMOTE                   211
*                        RESET: LOGOFF PROCESSING                   211
SCBEGRPC EQU   X'08'     SET: GROUP IN CONFIGURATION
```

```
SCBERSO  EQU   X'10'     SET: RSO DESTINATION                        211
SCBEDFER DC    X'00'         CONTROL SWITCH                         211
SCBEDRMT EQU   X'01'     SET: REMOTE DESTINATION                     211
*                        RESET: CENTRAL DESTINATION                  211
SCBEDFRR EQU   X'02'     SET: DEFERRED OUTPUT                        211
*                        RESET: IMMEDIATE OUTPUT                     211
SCBERESV EQU   X'04'     SET: RESERVED OUTPUT                        211
*                        RESET: NOT RESERVED OUTPUT                  211
SCBESUBM EQU   X'08'     SET: RESERVED BY SUBMITTER                  211
*                        RESET: RESERVED BY ALTERNATE                211
*-------------------------------------------------------------------*
*                     NEW FIELDS FOR SPOOL V2                        *
*-------------------------------------------------------------------*
SCBECH1C DS    H              # OF LINES BEF. CHANNEL1, CROSS HP
SCBETRLN DC    F'0'                    TRAILER NUMBER               211
SCBEHDRN DC    F'0'                    HEADER NUMBER                211
SCBEUSTX DC    CL32' '                 USER TEXT IN HEADER-TRAILER  213
SCBELIND DS    H              TOTAL LINES/PAGE FOR ND2
SCBELISD DS    H              TOTAL LINES/PAGE FOR SD23
SCBELILP DS    H              TOTAL LINES/PAGE FOR LP3337
SCBELIHP DS    H              TOTAL LINES/PAGE FOR HP (3351, 3353)
SCBEVBHP DS    H              # OF LINES FROM VFB FOR HP
SCBES2B  DS    X              SWITCH 2 BIS
SCBEHP   EQU   X'01'          JOB MAY BE DIRECTED ON HP
SCBEMHP  EQU   X'02'          JOB MUST BE DIRECTED ON HP
SCBECVFH EQU   X'04'          CHECK VFB REQUESTED FOR HP
SCBENDX  EQU   X'08'          JOB MAY BE DIRECTED ON HP54 (=NDX)
SCBEMNDX EQU   X'10'          JOB MUST BE DIRECTED ON HP54
SCBECVFX EQU   X'20'          CHECK VFB REQUESTED FOR HP54
SCBEMUNW EQU   X'40'          MUST NEW, FROM SPOOL 2.5
*
SCBERDT  DS    X              REQUESTED DEVICE TYPE
SCBEDTAL EQU   X'00'          *ALL
SCBEDTND EQU   X'01'          ND PRINTER
SCBEDTHP EQU   X'02'          HP PRINTER
*
SCBEHPFL DS    X              HP FLAG
SCBEFOBP EQU   X'01'          FOB POOL USED
SCBECHPO EQU   X'02'          CHARS POOL USED
SCBEROTF EQU   X'04'          ROTATION SPECIFIED IN REQUEST
SCBEH2SP EQU   X'08'          HP TO SPS JOB                           074
*
SCBELPHP DS    CL3            REQUESTED LOOP FOR HP
SCBELPND DS    CL3            REQUESTED LOOP FOR ND2
SCBELPSD DS    CL3            REQUESTED LOOP FOR SD23
SCBELPLP DS    CL3            REQUESTED LOOP FOR 3337
SCBEFDTY DS    X              FDTYPE PARAM
SCBEFSRA EQU   X'00'          SCRATCH
```

```
SCBE3170 EQU   X'40'          FD3170
SCBE1S1D EQU   X'F1'          FD1S1D (SINGLE SIDE, SINGLE DENSITY)
SCBE11BD EQU   X'40'          FD11BD (SINGLE SIDE, SINGLE DENSITY)
SCBE2S1D EQU   X'F2'          FD2S1D (DOUBLE SIDE, SINGLE DENSITY)
SCBE2S2D EQU   X'D4'          FD2S2D (DOUBLE SIDE, DOUBLE DENSITY)
*
SCBECH1H DS    H              # OF LINES BEFORE CHANEL 1 FOR HP
SCBEVFBH DS    H              SCBEVFBL FOR HP
SCBEC#DH DS    CL1            LPI AND CPI (CPI - 6) FOR HP FROM
*                             1RST CMD OR POOL CHARS OR DEFAULT CH.
SCBELIRD EQU   X'80'          RESERVED FOR CHARS# FROM FIRST CHARS
SCBELPI6 EQU   X'10'          6 LPI FOR HP
SCBELPI8 EQU   X'20'          8 LPI FOR HP
SCBELP10 EQU   X'30'          10 LPI FOR HP
SCBELP12 EQU   X'40'          12 LPI FOR HP
SCBELP24 EQU   X'50'          24 LPI FOR HP
SCBEHCPI EQU   X'0F'          CPI VALUE IN 4 LAST BITS
*
SCBEHPOS DS    CL4            $TSOS HPFILE CODED FILE ID
SCBEFIFL DS    X              FILE FLAG
SCBEFIPL EQU   1              PLAM LIB
SCBEPRER EQU   2              SECTION PREPROCESSING REQUESTED.
SCBEPREI EQU   4              SECTION IN PREPROCESS
SCBEPRES EQU   8              SECTION PREPROCESSED WITH SUCCESS
SCBEPREF EQU   16             PREPROCESSING FAILED WITH DMS ERROR
*                             (WHICH IS IN &P.SDMS).
SCBEPREL EQU   32             PREPROCESSING LOGICAL FAILURE
SCBECOFI EQU   64             FILE COPIES IS REQUESTED
SCBECOPA EQU   128            PAGE COPIES IS REQUESTED
*
SCBEPLRF DS    X              PLAM RECORD TYPE FROM
SCBEPLRT DS    X              PLAM RECORD TYPE TO
SCBEERRP DS    X              ERROR PROCESSING=
SCBETRIG EQU   1              TRUNC=IGNORE  | RESET =
SCBETRKE EQU   2              TRUNC=KEEP    |        STD
SCBEMGPY EQU   4              MSGPAGE=YES   | RESET =
SCBEMGPN EQU   8              MSGPAGE=NO    |        STD
*
SCBELP48 DS    CL3            REQUESTED LOOP FOR LP48
SCBELI48 DS    H              TOTAL LINES/PAGE FOR LP48
SCBEVB48 DS    H              # OF LINES FROM VFB FOR LP48
SCBELI65 DS    H              TOTAL LINES/PAGE FOR LP65 (=ND50)
SCBEVB65 DS    H              # OF LINES FROM VFB FOR LP65
SCBELIUL DS    H              TOTAL LINES/PAGE FOR EMULATED
SCBEVBUL DS    H              # OF LINES FROM VFB FOR EMULATED
SCBELP65 DS    CL3            REQUESTED LOOP FOR LP65
SCBELPUL DS    CL3            REQUESTED LOOP FOR EMULATED PRINTER
SCBEDSEM DS    H              DISPLACEMENT TO DSEM PRINT PARAMETER
```

```
SCBEFACR DS    X             NUMBER OF FILES IN A FAMILY(1RST MEM.)
SCBESEFL DS    X             SECTION FLAG
SCBESERE EQU   1             SECTION REQUESTED IN THE COMMAND
SCBESEBF EQU   2             FROM BEGIN OF FILE
SCBESEFY EQU   4             FROM *ANY
SCBESEEF EQU   8             TO END OF FILE
SCBESETY EQU   16            TO *ANY
SCBESCTY EQU   32            SET = SECT  IN CHR, RESET = HEXA
SCBEFITY EQU   64            SET = FIRST IN CHR, RESET = HEXA
SCBELATY EQU   128           SET = LAST  IN CHR, RESET = HEXA
*-------------------------------------------------------------------*
*                         CHARS FOR RSO                             *
*-------------------------------------------------------------------*
SCBEFT5  DS    XL3           FONT 5 KEY (SRC SEQ # MODIF)
SCBEFT6  DS    XL3           FONT 6 KEY
SCBEFT7  DS    XL3                     FONT 7 KEY                  211
SCBEFT8  DS    XL3                     FONT 8 KEY                  211
SCBEFT9  DS    XL3                     FONT 9 KEY                  211
SCBEFT10 DS    XL3                     FONT 10 KEY                 211
SCBEFT11 DS    XL3                     FONT 11 KEY                 211
SCBEFT12 DS    XL3                     FONT 12 KEY                 211
SCBEFT13 DS    XL3                     FONT 13 KEY                 211
SCBEFT14 DS    XL3                     FONT 14 KEY                 211
SCBEFT15 DS    XL3                     FONT 15 KEY                 211
SCBEFT16 DS    XL3                     FONT 16 KEY                 211
SCBEFTTL EQU   *-SCBEFT5+12     TOTAL FONTS LENGTH                  211
SCBEFT#  EQU   SCBEFTTL/L'SCBEFT1   FONTS NUMBER                    211
SCBEC#D1 DS    CL(SCBEFT#)  DENSITIES CORR. TO (16) FONTS          211
*-------------------------------------------------------------------*
*                 GENERAL OUTPUT INFORMATIONS                       *
*-------------------------------------------------------------------*
SCBETSPO DC    XL3'000'          TIME JOB SPOOLED OUT HHMMSS       211
SCBETSPF DC    XL3'000'          TIME JOB SPOOLED OUT FINISH HHMMSS 211
SCBELIHC DS    H             TOTAL LINES/CROSS PAGE FOR HP
SCBELPP# DS    F             LAST RETRIEVAL @: BLOCK#(BBBBBBFF)    SBC
SCBE#PP  DS    F             FILE SIZE IN PRINTER PAGES            SBC
SCBECH1# DS    F             COUNTER OF SKIP TO CHANNEL 1          SBC
SCBESBC# DS    H             PRINT SEQUENCE #                      SBC
*-------------------------------------------------------------------*
*                   RSO 2.0 SPECIFIC FIELDS                         *
*-------------------------------------------------------------------*
SCBETRA# DS    X             INPUT TRAY # FOR RSO & NDY
SCBEITMA EQU   X'80'                   = MANUAL                    011
SCBEITA3 EQU   X'81'                   = A3                        011
SCBEITA4 EQU   X'82'                   = A4                        011
SCBEITA5 EQU   X'83'                   = A5                        011
SCBEITB4 EQU   X'84'                   = B4                        011
SCBEITB5 EQU   X'85'                   = B5                        011
```

```
SCBEITDO EQU   X'86'                    = DOUBLE-LETTER          011
SCBEITEX EQU   X'87'                    = EXEC                   011
SCBEITFO EQU   X'88'                    = FOLIO                  011
SCBEITIV EQU   X'89'                    = INVOICE                011
SCBEITLG EQU   X'8A'                    = LEGAL                  011
SCBEITLT EQU   X'8B'                    = LETTER                 011
SCBEITMO EQU   X'8C'                    = MONARCH                011
SCBEIT10 EQU   X'8D'                    = COM10                  011
SCBEITDL EQU   X'8E'                    = DL                     011
SCBEITC5 EQU   X'8F'                    = C5                     011
SCBEITIG EQU   X'FF'                    = IGNORE                 011
*
SCBESQSA DS    X            PREVIOUS "SQ" KIND SAVED
SCBEXSEC DS    F            TDFILE SECTION NUMBER
SCBESHRO DS    D            RSO SCHEDULING
SCBETRAT DS    CL8          TRANSLATION TABLE NAME
SCBERDST DS    CL8          RSO AND LOCAL DESTINATION GROUP
SCBEPINF DS    X            RSO PRINT INFO
SCBECHMO EQU   X'01'        CHARS MODIFICATION
SCBERSMP EQU   X'02'        SET: SAMPLE=YES GIVEN
SCBERDI  EQU   X'04'        SET: RDIR=NO
*
SCBEXTYP DS    X            MODE TYPE
SCBEXRSO EQU   X'00'        XTYPE=RSO (DEFAULT)
SCBEXRTD EQU   X'01'        XTYPE=TDSPOOL
*
SCBESPC2 DS    X            SPACE OPERAND FOR RSO 2.0
SCBESPE  EQU   X'00'        SPACE=E
SCBESP1  EQU   X'01'        SPACE=1
SCBESP2  EQU   X'02'        SPACE=2
SCBESP3  EQU   X'03'        SPACE=3
SCBESPN  EQU   X'04'        SPACE=NO
*
SCBETYP3 DS    X            SPACE OPERAND FOR RSO 2.0
SCBESPI  EQU   X'01'        SPACE=IBM
SCBESPA  EQU   X'02'        SPACE=ASA
SCBELN2  EQU   X'04'        LINES=SPECIFIED
SCBESPTE EQU   X'80'        SPACE=TE
SCBESPTI EQU   X'81'        SPACE=TI
SCBESPTA EQU   X'82'        SPACE=TA
*
SCBECTL2 DS    X            CONTROL OPERAND FOR RSO 2.0
SCBECTNO EQU   X'01'        CONTROL=NO
SCBECTPH EQU   X'02'        CONTROL=PHYSICAL
SCBECTLG EQU   X'04'        CONTROL=LOGICAL
SCBECTLN EQU   X'10'        CONTROL=LINEMODE
SCBECTTR EQU   X'20'        CONTROL=TRANSPARENT
*
```

```
SCBETTIM DS    CL4           IMAGE FOR TRANSLATION TABLE
SCBEEXHC DS    CL8           EXTENDED HOST CODE
SCBESRMD DS    X             SORT MODE                                  011
SCBESMNO EQU   X'01'               = NO                                011
SCBESMGR EQU   X'02'               = GROUP                             011
SCBESMCO EQU   X'03'               = COLLATE                           011
SCBESMST EQU   X'04'               = STACKER                           011
SCBESMAU EQU   X'05'               = AUTOMATIC                         011
*
SCBERPPD DS    H             RSO PRINT PARAM. DISPLACEMENT
SCBEESTS DS    F             ESTIMATE SIZE FOR STATUS
SCBETOOS DS    H             TOP OFFSET                                 011
SCBELEOS DS    H             LEFT OFFSET                                011
SCBEMTSN DS    CL4           OLD TSN DURING /MOVE-PR-JOB COMMAND        077
SCBEMOVE DS    X             INDICATOR FOR /MOVE-PR-JOB COMMAND         077
SCBENEW  EQU   X'80'          NEW TSN HAS BEEN ASSIGNED DURING MOVE     077
SCBEMVPJ EQU   X'40'          MOVE-PR-JOB COMMAND GIVEN                 077
SCBEMPJA EQU   X'20'          MODIFY-PR-JOB-ATTR COMMAND GIVEN          077
SCBENSCH EQU   X'10'          JOB IS NOT SCHEDULABLE                    077
         DS    CL1           RESERVED FOR RSO 2.0
*-------------------------------------------------------------------------*
*                      GENERAL INFORMATIONS                               *
*-------------------------------------------------------------------------*
SCBECPUT DS    XL8           CPU TIME USED BY SPOOL
SCBEUTAP DS    XL8           USER PARAM TAPE=
         ORG   SCBEUTAP
SCBENFNM DS    CL54          NEW SAM FILENAME
SCBEFUT  DS    0F            UNUSED IN VER = 102                        102
         ORG   SCBETOP+X'3F8'                                           102
*-------------------------------------------------------------------------*
*  ATTENTION:  THE REAL LENGTH OF THE SCB IS IN &P.TOP. !!!!!!!!!!!   *
*-------------------------------------------------------------------------*
SCBELIEX DS    CL1024        MAXIMUM EXTENSION LENGTH
         ORG   SCBELIEX    FIXED PART OF THE EXTENSION
SCBEPLRA DS    F           MAXIMUM RECSIZE FOR SATZART A
SCBEPLRB DS    F           MAXIMUM RECSIZE FOR SATZART B
SCBEPLED DS    H           DISPLACEMENT FOR PLAM ELEMENT NAME
*                             FROM THE BEGIN OF THE COMPLETE SCB.
SCBEPLVD DS    H           IDEM FOR PLAM ELEMENT VERSION.
SCBEPLTD DS    H           IDEM FOR PLAM ELEMENT TYPE
SCBEJVDI DS    H           IDEM FOR JOB VARIABLE
SCBEJPWD DS    CL4         USER JOB VARIABLE PASSWORD
SCBECPY# DS    H           NUMBER OF PRINT EXEMPLARS
SCBEPARM DS    H           DISPL. TO PRINT PARAMETERS
*
SCBEJVEX DS    X           JOB VARIABLE INDICATOR
SCBEJVXY EQU   1           JV EXISTS
SCBEJVXN EQU   2           JV INACCESSIBLE
```

```
SCBEJUWP EQU   4          USER PASSWORD TO RESTORE
SCBEJVCR EQU   8          JV CREATED BY SPOOL
*
SCBESDMS DS    X          SECTION DMS ERROR CODE INDICATOR
SCBESDEF EQU   1          EMPTY FILE
SCBESDOP EQU   2          DMS CODE IN SCBEPERR
SCBESDSE EQU   4          SETL ERROR
*
SCBEPERR DS    CL4        DMS ERROR CODE OF PREPROCESSING
SCBEDIAR DS    X          DIA RECTO FOR LP65 (=ND50)
SCBEDIAV DS    X          DIA VERSO FOR LP65
SCBESIDD DS    H          DISPLACEMENT FOR SECTION ID
SCBESEPO DS    H          SECTION ID POSITION
SCBESF#  DS    F          SECTION FROM NUMBER
SCBESFDI DS    H          DISPLACEMENT OF FROM SECTION ID
SCBESFPO DS    H          SECTION FROM POSITION
SCBEST#  DS    F          SECTION TO   NUMBER
SCBESTDI DS    H          DISPLACEMENT OF TO   SECTION ID
SCBESTPO DS    H          SECTION TO   POSITION
SCBESLB# DS    F          SETL BLOCK NUMBER
SCBESLR# DS    F          SETL RECORD NUMBER
SCBER#BF DS    F          RECORD NUMBER FROM BEGIN OF FILE
SCBEFSCR DS    F          FROM (FIRST) STRING COUNTER
SCBETSCR DS    F          TO (LAST) STRING COUNTER
SCBESSCR DS    F          SECTION STRING COUNTER
SCBEISKY DS    H          ISAM KEY DISPLACEMENT FOR SECTIONS
SCBEIMDI DS    H          DISPLACEMENT FOR IMAGE FILENAME
*
SCBECKPT DS    X          CHEKPOINT TO DO
SCBECKPP EQU   0                        AT PAGE
SCBECKPS EQU   1                        AT SECTION
*
SCBEDPLX DS    X          DUPLEX =
SCBEDUPD EQU   0                    STD
SCBEDUPY EQU   1                    YES
SCBEDUPT EQU   2                    TUMBLE
SCBEDUPN EQU   4                    NO
*
SCBEPACC DS    X          PAGE CONTROL CHARACTER (TOP OF PAGE)
SCBEPACS EQU   0          STD
SCBEPACN EQU   1          NO
*
SCBEOTRA DS    X          OUTPUT TRAY # LP65 (=ND50,NDY)
SCBEOTIG EQU   X'80'      OUT-TRAY = NO      ( RSO )              011
SCBEOTSO EQU   X'81'      OUT-TRAY = SORTER  ( RSO )              011
*
SCBENPAD DS    F          PAGEDEF VALUE FOR LP65
SCBEPCCD DS    F          PRINT CONTROL CHARACTER POSITION
```

```
SCBETTID DS    H          TRANSLATION TABLE IMAGE DISPL.
SCBEVBNX DS    H          USABLE LINES FROM VFB (HP54=NDX)
SCBELINX DS    H          TOTAL LINES#/PAGE (HP54)
SCBECH1X DS    H          LINES# BEFORE CHANNEL 1 (HP54)
SCBELPNX DS    CL3        REQUESTED LOOP FOR HP54
SCBELXI  DS    X          LOOP LENGTH IN 1/6 INCH (HP54)
SCBERPMX DS    F          RASTER PATTERN MEMORY IN KB (HP54)
SCBETWUX DS    X          TWO-UP MODE FRO: RECORD FORM
SCBETW1X EQU   X'01'      MODE 1
SCBETW2X EQU   X'02'      MODE 2
SCBETW3X EQU   X'04'      MODE 3
SCBETWNX EQU   X'80'      NO TWO-UP
*
SCBEC#DX DS    CL1        LPI AND CPI (CPI-6) FOR HP54 FROM 1RST
*                         CMD OR POOL OR DEFAULT CHARS
*                         TO USE WITH THE EQU OF SCBOC#DH.
SCBEITRA DS    X          INPUT  TRAY # (LP65=NDY,ND50)
SCBERESR DS    X          RESOURCE PRINTER
SCBEND65 EQU   X'80'      LP65
SCBED65R EQU   X'40'      DIA RECTO FOR LP65 SPECIFIED
SCBED65V EQU   X'20'      DIA VERSO FOR LP65 SPECIFIED
SCBERAPA EQU   X'10'      APA
SCBERFOD EQU   X'08'      FORMDEF SPECIFIED FOR APA
SCBERPAD EQU   X'04'      PAGEDEF SPECIFIEC FOR APA
SCBETRCY EQU   X'02'      TAB-REF-CHAR = YES
SCBELIP  EQU   X'01'      LIP-2090                                    002
*
SCBEVBXC DS    H          USABLE LINES FROM CROSS VFB (HP54=NDX)
SCBELIXC DS    H          CROSS TOTAL LINES#/PAGE (HP54)
SCBECX1C DS    H          CROSS LINES# BEFORE CHANNEL 1 (HP54)
SCBECRLX DS    CL3        REQUESTED CROSS LOOP FOR HP54
SCBELXIC DS    X          CROSS LOOP LENGTH IN 1/6 INCH (HP54)
SCBETRCD DS    H          DISPLACEMENT TO TABLE REFERENCE CHAR
*                         [A(SCB) + &P.TRCD] POINTS TO AN AREA
*                         OF (4 * 8) BYTES.
SCBETSAV DS    F          TO (LAST) STRING COUNTER SAVED
SCBESSAV DS    F          SECTION COUNTER SAVED
SCBE50FO DS    CL8        FORMDEF FOR APA-2050
SCBE50PA DS    CL8        PAGEDEF FOR APA-2050
SCBE90FO DS    CL8        FORMDEF FOR APA-2090
SCBE90PA DS    CL8        PAGEDEF FOR APA-2090
SCBEKPOS DS    H          KEYPOS OF FILE TO BE PRINTED
SCBEFKLN DS    X          KEYLEN OF FILE TO BE PRINTED
SCBELAGX DS    X          LANGUAGE EXTENSION
SCBEARTL EQU   X'01'         = ARABIC RIGHT TO LEFT
SCBEALTR EQU   X'02'         = ARABIC LEFT TO RIGHT
SCBEFRTL EQU   X'03'         = FARSI RIGHT TO LEFT
SCBEFLTR EQU   X'04'         = FARSI LEFT TO RIGHT
```

```
*
SCBEFOND DS    H          DISPLACEMENT TO FORMAT NAME
SCBETMBS DS    C          START TIME JOB SPOOLED OUT (SUMMER/WINTER)
SCBETMES DS    C          END TIME JOB SPOOLED OUT (SUMMER/WINTER)
SCBELOTS DS    C          LOGOFF TIME (SUMMER/WINTER)
SCBEDOCF DS    X          DOCUMENT-FORMAT                          001
SCBEDCFT EQU   X'00'         = TEXT
SCBEDCFP EQU   X'01'         = PAGE-FORMAT
SCBEDCFS EQU   X'02'         = SPECIAL-FORMAT
*
SCBEUFND DS    H          DISPLACEMENT TO UFS FILENAME
SCBEUCND DS    H          DISPLACEMENT TO UFS COPY NAME
SCBECFRM DS    CL6        CLIENT FORM FOR EXIT 94
SCBEMLSV DS    H          MAX LINES OF PRINT REQUEST
SCBEDIFD DS    H          DISPLACEMENT TO DIA-IN-FORM STRUCTURE    002
SCBESPSD DS    H          DISPLACEMENT TO SPS PRINT PARAMETER      022
SCBEDPRD DS    H          DISPLACEMENT TO DPRINT PRINT PARAMETER   022
SCBEFLTD DS    H          DISPLACEMENT TO FILTER WORK FILE INFORM. 022
SCBEFRMD DS    H          DISPLACEMENT TO DOCUMENT FORMAT          022
SCBESEND DS    F          END OF SECTION (# RECORDS)               022
SCBECMXI DS    0XL20      CMX_INFORMATION
SCBECMXH DS    F              HEADER
SCBECMXN DS    CL8            HOST NAME
SCBECMXA DS    CL8            APPLICATION
SCBEOHST DS    CL8        ORIGINAL HOST NAME
SCBECTSN DS    CL4        ORIGINAL TSN
SCBEIHST DS    CL8        ICRM HOST NAME
SCBESERV DS    CL8        ORIGINAL SERVER NAME
SCBEDHST DS    CL8        DESTINATION HOST NAME
SCBEDTSN DS    CL4        SERVER TSN
SCBECIFT DS    CL4        CATID OF FILE TO BE TRANSFER
SCBECICT DS    CL4        CATID OF CONTAINER
SCBERDSB DS    CL8        ORIGINAL SCHEDULING BITS FOR RDIR
SCBEVIDS DS    H          DISPLACEMENT TO PROGRAM NAME AND STRING  021
SCBEAPSC DS    H          DISPLACEMENT TO SPS SCHEDULING AREA      063
SCBEPCLD DS    H          DISPLACEMENT TO PCL INFORMATION STRUCTURE 071
SCBEAIDD DS    H          DISPLACEMENT TO ACCOUNT-ID               073
SCBESDAT DS    F          SCHEDULING DATE                          074
SCBESTIM DS    H          SCHEDULING TIME                          074
         ORG   SCBESDAT
SCBESCDC DS    CL6        DATE AND TIME IN CHAR                    074
         ORG   SCBESDAT
SCBESCDB DS    XL6        DATE AND TIME IN HEXA                    074
SCBELUID DS    H          DISPLACEMET TO LONG USER-ID              076
SCBEAPAD DS    H          APA OVERLAY DISPLACMENT                  078
         DS    H          DISPL FOR NEW EXTENSIONS - DON'T TOUCH !  078
SCBEVARI DS    CL728      VARIABLE PART OF THE EXTENSION
SCBEVAR2 DS    CL1024     EXTENSION FOR POSIX FILENAME
```

```
*************************************************************************
*                                                                       *
*        THE VARIABLE PART MUST BE USED FOR PARAMETERS OF VARIABLE      *
*        LENGTH, I.E. PLAMLIB ELEMENT NAME, JV NAME, SECTION ID.        *
*        EACH VARIABLE NAME MUST BE PRECEEDED BY ITS LENGTH (1 BYTE).   *
*        THE ADDRESS OF THIS FIELD = SCB @ + SCB DISPLACEMENT FIELD.    *
*        THE DISPLACEMENT FIELDS ARE EITHER IN THE FIXED PART OR IN     *
*        THE EXTENSION FIXED PART OF THE SCB.                           *
*        FOR EXAMPLE :                                                  *
* &P.PLEL  DS    X        LENGTH OF PLAM ELEMENT NAME             930 *
* &P.PLEN  DS    CL(&P.PLEL)  PLAM ELEMENT NAME                   930 *
*        THE ADDRESS OF &P.PLEL = ADDR(SCB) + SCBOPLED.                 *
*                                                                       *
*************************************************************************
SCBESLEN EQU   *-SCBESCB
SCBELEN  EQU   *-SCBETOP
         END
```

6.2 SVC lists

6.2.1 List of SVCs valid for system exits

Sorted according to SVC numbers:

SVC number decimal	SVC number hexa.	Macro name (valid for system exits)	Comments (table sorted acc. to SVC number)
1	1	RELM (31-bit)	
1	1	REQM (31-bit)	
1	1	DISMP (31-bit)	
1	1	ENAMP (31-bit)	
1	1	RELMP (31-bit)	
1	1	REQMP (31-bit)	
1	1	MINF (31-bit)	
1	1	CSTMP (31-bit)	
1	1	CSTAT (31-bit)	
2	2	LPOV	
7	7	DCLVAR	
7	7	DELVAR	
7	7	SETVAR	
7	7	GETVAR	
7	7	GETNVAR	
7	7	SETNVAR	
7	7	POSVAR	
23	17	GETOD	
25	19	DUMP (24-bit)	
26	1A	CDUMP (31-bit)	
27	1B	CDUMP	
28	1C	QUIET	
30	1E	EXCALL	
31	1F	EXECITC	
32	20	DELPOOL	cf. notes below
32	20	CREAPOOL	cf. notes below
33	21	STAMCE	
35	23	GEPRT	
36	24	BCAM-SVC	
40	28	EAM (31-bit)	
42	2A	SWITCH	
43	29	MAREN	LOGON environment required
45	2D	FITC	
46	2E	ADAM	
49	31	GETUGR	
49	31	SRMSUG	
49	31	RDUID	

SVC number decimal	SVC number hexa.	Macro name (valid for system exits)	Comments (table sorted acc. to SVC number)
56	38	XHCS	
64	40	SETUS	
65	41	GETUS	
70	46	TMODE	
70	46	TSTAT	
70	46	TCHNG	
71	47	SETSW	
72	48	GETSW	
76	4C	PASS	
79	4F	SVC79	
84	54	PFMON	only on special release
87	57	GTMAP	
89	59	VPASS	
95	5F	AUDIT	
96	60	MSG7	
99	63	AINF	
99	63	AREC	
99	63	ASPC	
100	64	HOSTPR	
102	66	NKGTYPE	
103	67	VMGINF	
104	68	ISREQ	
105	69	OSTAT	
117	75	VSVI	cf. notes below
117	75	PBUNLD	cf. notes below
119	77	DSSMTST1	
120	78	DSSMTST2	
127	7F	MRSINF	
129	81	AIDTRAP	
130	82	TAM	
135	87	SINF	
135	87	TINF	
138	8A	DHSRCL	
140	8C	JOBINFO	
140	8C	JOBINFOL	
140	8C	JOBINFOS	
144	90	FSTAT (31-bit)	
144	90	RDTFT (31-bit)	
144	90	ERASE (31-bit)	
144	90	CATAL (31-bit)	
144	90	IMPORT (31-bit)	
144	90	FILE (31-bit)	
146	92	GEPRT (31-bit)	
148	94	$ITRM	
149	95	$IGPT	

SVC number decimal	SVC number hexa.	Macro name (valid for system exits)	Comments (table sorted acc. to SVC number)
150	96	TYPIO	
168	A8	ELIM (31-bit)	cf. notes below
168	A8	GET (31-bit)	cf. notes below
168	A8	GETFL (31-bit)	cf. notes below
168	A8	GETKY (31-bit)	cf. notes below
168	A8	GETR (31-bit)	cf. notes below
168	A8	GOTO (31-bit)	cf. notes below
168	A8	INSRT (31-bit)	cf. notes below
168	A8	ISREQ (31-bit)	cf. notes below
168	A8	OSTAT (31-bit)	cf. notes below
168	A8	PUT (31-bit)	cf. notes below
168	A8	PUTX (31-bit)	cf. notes below
168	A8	RETRY (31-bit)	cf. notes below
168	A8	SETL (31-bit)	cf. notes below
168	A8	STORE (31-bit)	cf. notes below
169	A9	RELSE (24-bit)	
170	AA	PAM (31-bit)	
171	AB	BTAM (31-bit)	
172	AC	IOSID	
177	B1	EAM (24-bit)	cf. notes below
179	B3	PAM (24-bit)	cf. notes below
181	B5	CHNGE	
182	B6	COPY	
183	B7	LINK (31-bit)	cf. notes below
183	B7	TABLE (31-bit)	cf. notes below
183	B7	ITABL (31-bit)	cf. notes below
183	B7	SULNK (31-bit)	cf. notes below
186	BA	LBRET (31-bit)	
186	BA	OPEN (31-bit)	
186	BA	CLOSE (31-bit)	
186	BA	EXRTN (31-bit)	
186	BA	FEOV(31-bit)	
188	BC	GET (31-bit)	
188	BC	PUT (31-bit)	
188	BC	SETL (31-bit)	
188	BC	PUTX (31-bit)	
188	BE	RELSE (31-bit)	
190	BE	GETJV (31-bit)	cf. notes below
190	BE	SETJV (31-bit)	cf. notes below
190	BF	STAJV (31-bit)	cf. notes below
191	BF	JSATTCH (31-bit)	
191	BF	JSDETCH (31-bit)	
191	BF	JSEXPCT (31-bit)	
191	BF	JSINFO (31-bit)	
191	BF	JSWAKE (31-bit)	

SVC number decimal	SVC number hexa.	Macro name (valid for system exits)	Comments (table sorted acc. to SVC number)
191	BF	JSRUNJB (31-bit)	
191	BF	JINF (31-bit)	
191	CE	TMODE (31-bit)	
206	CE	§NKACALL (31-bit)	
208	D0	RDUID	
209	D1	$DALL	
221	DD	$VRIF	
228	E4	$REQM	
229	E5	$RELM	
237	ED	$TERM	

Sorted according to macro names

SVC number decimal	SVC number hexa.	Macro name (valid in system exits)	Comments (table sorted acc. to macro name)
149	95	$IGPT	
148	94	$ITRM	
206	CE	$NKACALL (31-bit)	
229	E5	$RELM	
228	E4	$REQM	
237	ED	$TERM	
221	DD	$VRIF	
46	2E	ADAM	
129	81	AIDTRAP	
99	63	AINF	
99	63	AREC	
99	63	ASPC	
95	5F	AUDIT	
36	24	BCAM-SVC	
171	AB	BTAM (31-bit)	
144	90	CATAL (31-bit)	
26	1A	CDUMP (31-bit)	
27	1B	CDUMP	
181	B5	CHNGE	
186	BA	FEOV(31-bit)	
182	B6	COPY	
32	20	CREAPOOL	cf. notes below
1	1	CSTAT (31-bit)	
1	1	CSTMP (31-bit)	
7	7	DCLVAR	
32	20	DELPOOL	cf. notes below
7	7	DELVAR	
138	8A	DHSRCL	
1	1	DISMP (31-bit)	
119	77	DSSMTST1	
120	78	DSSMTST2	
25	19	DUMP (24-bit)	
40	28	EAM (31-bit)	
177	B1	EAM (24-bit)	cf. notes below
168	A8	ELIM (31-bit)	cf. notes below
1	1	ENAMP (31-bit)	
144	90	ERASE (31-bit)	
30	1E	EXCALL	
31	1F	EXECITC	
186	BA	EXRTN (31-bit)	
186	BA	FEOV(31-bit)	
144	90	FILE (31-bit)	
45	2D	FITC	

SVC number decimal	SVC number hexa.	Macro name (valid in system exits)	Comments (table sorted acc. to macro name)
144	90	FSTAT (31-bit)	
35	23	GEPRT	
146	92	GEPRT (31-bit)	
168	A8	GET (31-bit)	cf. notes below
188	BC	GET (31-bit)	
168	A8	GETFL (31-bit)	cf. notes below
190	BE	GETJV (31-bit)	cf. notes below
168	A8	GETKY (31-bit)	cf. notes below
7	7	GETNVAR	
23	17	GETOD	
168	A8	GETR (31-bit)	cf. notes below
72	48	GETSW	
49	31	GETUGR	
65	41	GETUS	
7	7	GETVAR	
168	A8	GOTO (31-bit)	cf. notes below
87	57	GTMAP	
100	64	HOSTPR	
144	90	IMPORT (31-bit)	
168	A8	INSRT (31-bit)	cf. notes below
172	AC	IOSID	
104	68	ISREQ	
168	A8	ISREQ (31-bit)	cf. notes below
183	B7	ITABL (31-bit)	cf. notes below
191	BF	JINF (31-bit)	
140	8C	JOBINFO	
140	8C	JOBINFOL	
140	8C	JOBINFOS	
191	BF	JSATTCH (31-bit)	
191	BF	JSDETCH (31-bit)	
191	BF	JSEXPCT (31-bit)	
191	BF	JSINFO (31-bit)	
191	BF	JSRUNJB (31-bit)	
191	BF	JSWAKE (31-bit)	
186	BA	LBRET (31-bit)	
183	B7	LINK (31-bit)	cf. notes below
2	2	LPOV	
43	29	MAREN	LOGON enviroment required
1	1	MINF (31-bit)	
127	7F	MRSINF	
96	60	MSG7	
102	66	NKGTYPE	
186	BA	OPEN (31-bit)	
105	69	OSTAT	
168	A8	OSTAT (31-bit)	cf. notes below

SVC number decimal	SVC number hexa.	Macro name (valid in system exits)	Comments (table sorted acc. to macro name)
170	AA	PAM (31-bit)	
179	B3	PAM (24-bit)	cf. notes below
76	4C	PASS	
117	75	PBUNLD	cf. notes below
84	54	PFMON	only on special release
7	7	POSVAR	
168	A8	PUT (31-bit)	cf. notes below
188	BC	PUT (31-bit)	
168	A8	PUTX (31-bit)	cf. notes below
188	BC	PUTX (31-bit)	
28	1C	QUIET	
144	90	RDTFT (31-bit)	
49	31	RDUID	
208	D0	RDUID	
1	1	RELM (31-bit)	
1	1	RELMP (31-bit)	
169	A9	RELSE (24-bit)	
188	BE	RELSE (31-bit)	
1	1	REQM (31-bit)	
1	1	REQMP (31-bit)	
168	A8	RETRY (31-bit)	cf. notes below
190	BE	SETJV (31-bit)	cf. notes below
168	A8	SETL (31-bit)	cf. notes below
188	BC	SETL (31-bit)	
7	7	SETNVAR	
71	47	SETSW	
64	40	SETUS	
7	7	SETVAR	
135	87	SINF	
49	31	SRMSUG	
190	BF	STAJV (31-bit)	cf. notes below
33	21	STAMCE	
168	A8	STORE (31-bit)	cf. notes below
183	B7	SULNK (31-bit)	cf. notes below
79	4F	SVC79	
42	2A	SWITCH	
183	B7	TABLE (31-bit)	cf. notes below
130	82	TAM	
70	46	TCHNG	
135	87	TINF	
70	46	TMODE	
191	CE	TMODE (31-bit)	
70	46	TSTAT	
150	96	TYPIO	
103	67	VMGINF	

SVC number decimal	SVC number hexa.	Macro name (valid in system exits)	Comments (table sorted acc. to macro name)
89	59	VPASS	
117	75	VSVI	cf. notes below
56	38	XHCS	

6.2.2 List of SVCs not to be used in system exits

Sorted according to SVC numbers

SVC number decimal	SVC number hexa.	Macro name (not to be used in system exits)	Comments (table sorted acc. to SVC number)
3	3	NPCSVC	
5	5	WRCPT (31-bit)	
6	6	Meß-SVC	
8	8	SRMASVC	
9	9	TERM	Use $TERM
10	A	NCAN	
11	B	IOTRACE	
12	C	NPCHK	
13	D	VMMUSER	
14	E	NKDINF	
15	F	IPPMBASE	
16	10	CRYPT	
22	16	TERMD	
24	18	GEPRT (24-bit)	
29	1D	SCA	
36	24	BCAM-SVC	
37	25	REQSPF	
38	26	MSG (31-bit)	
39	27	RDCRD	cf. notes below
39	27	RDATA	cf. notes below
39	27	WROUT	cf. notes below
39	27	WRLST	cf. notes below
39	27	WRTRD	cf. notes below
39	27	WRTOT	cf. notes below
39	27	$EOUT	cf. notes below
41	29	SAT	
44	2C	YTCMAC	
47	2F	PLAM	
48	30	WARTOPT	
50	32	OPCOM	
51	33	SEVNT	
52	34	REVNT	
53	35	RELBF	
54	36	CLCOM	
57	39	CSSCN	
59	3B	MONTB	
60	3C	ERFLG	
61	3D	WRTOT	
62	3E	RDCRD	
63	3F	TERMJ	
63	3F	TERMJD	

SVC number decimal	SVC number hexa.	Macro name (not to be used in system exits)	Comments (table sorted acc. to SVC number)
66	42	RDATA	
67	43	WROUT	
68	44	WRTRD	
69	45	WRLST	
73	49	REQM (24-bit)	
74	4A	RELM (24-bit)	
75	4B	CSTAT (24-bit)	
81	51	JSINF (24-bit)	
81	51	JSINFO (24-bit)	
81	51	JSRUNDB (24-bit)	
81	51	JSFXCPT (24-bit)	
81	51	JSDETCH (24-bit)	
81	51	JSATTCH (24-bit)	
83	53	SM2	Usable only by SM2
85	55	IC02000	
86	56	SETBF	
88	58	CMD (24-bit)	
92	5C	BKPT	
93	5D	MSG (24-bit)	
94	5E	FASTACC	
94	5E	FASTPAM	
98	62	DBSVC	
106	6A	UNLOD	
107	6B	PRNT	
108	6C	$SPRQ	
108	6C	PNCH	
109	6D	ITABL (24-bit)	
110	6E	LINK (24-bit)	
110	6E	SULNK	
111	6F	TABLE	
112	70	VOLIN	
113	71	INITMB	
114	72	SFDLOAD	
116	74	NLMSERVE	
118	76	$SPORQ (RSO)	
121	79	DISSI	Serialization
121	79	ENASI	Serialization
121	79	CHKSI	Serialization
121	79	DEQAR	Serialization
121	79	ENQAR	Serialization
122	7A	ENAMP (24-bit)	
122	7A	REQMP (24-bit)	
122	7A	RELMP (24-bit)	
122	7A	DISMP (24-bit)	
123	7B	SUSPEND	Contingency

SVC number decimal	SVC number hexa.	Macro name (not to be used in system exits)	Comments (table sorted acc. to SVC number)
123	7B	LEVCO	Contingency
123	7B	RETCO	Contingency
123	7B	DISCO	Contingency
123	7B	ENACO	Contingency
124	7C	POSSIG	
124	7C	SOLSIG	Use $EXDEV, $EXEVI, EEXPOS (eventing)
124	7C	ENAEI	
124	7C	CHKEI	
124	7C	DISEI	
125	7D	(DCAM calls)	
126	7E	DIV	
128	80	STXIT	
128	80	CONTXT	
128	80	EXIT	
131	83	ARCHIVE	
132	84	SPOOLFREE	
133	85	STAM (24-bit)	
133	85	ONEVT (24-bit)	
133	85	STAJV (24-bit)	
133	85	DCLJV (24-bit)	
133	85	CATJV (24-bit)	
133	85	ERAJV (24-bit)	
133	85	GETJV (24-bit)	
133	85	SETJV (24-bit)	
134	86	KDCEND	
139	8B	SIR	
141	8D	IOCFCOPY	
142	8E	LOADPM	
143	8F	FDDRL	
145	91	CMD (31-bit)	
147	93	FEOV (24-bit)	
151	97	GET (24-bit)	
152	98	PUT (24-bit)	
153	99	PUTX (24-bit)	
154	9A	SETL (24-bit)	
155	9B	STORE (24-bit)	
156	9C	OPEN (24-bit)	
157	9D	CATAL (24-bit)	
158	9E	ERASE (24-bit)	
159	9F	FILE (24-bit)	
160	A0	FSTAT (24-bit)	
160	A0	RDTFT (24-bit)	
161	A1	RELEASE (24-bit)	
162	A2	CLOSE (24-bit)	
163	A3	GOTO (24-bit)	

SVC number decimal	SVC number hexa.	Macro name (not to be used in system exits)	Comments (table sorted acc. to SVC number)
164	A4	GETFL (24-bit)	
165	A5	VERIF (24-bit)	
166	A6	IMPORT (24-bit)	
173	AD	INSRT (24-bit)	
174	AE	ELIM (24-bit)	
175	AF	GETKY (24-bit)	
176	B0	GETR (24-bit)	
178	B2	EXRTN (24-bit)	
178	B2	LBRET (24-bit)	
180	B4	BTAM (24-bit)	
184	B8	POSIX	
185	B9	SRMUINF	
187	BB	RSOFEI	Eventing
187	BB	DELFEI	Eventing
187	BB	DSOFEI	Eventing
187	BB	RPOFEI	Eventing
187	BB	DPOFEI	Eventing
189	BD	FT-BS2000	
199	C7	$DOER	
213	D5	$EXCP	
214	D6	$WAIT	
215	D7	$XCPW	
216	D8	$PGFX	
216	D8	$PGFR	
222	DE	$AUDSW	
233	E9	$FNAT	
234	EA	$FNDT	
235	EB	$PEND	
236	EC	$UNPN	
238	EE	$PNUP	
241	F1	$PNDT	
245	F5	$MASK	
246	F6	$UNMASK	
250	FA	$BOWT	
255	FF	DIAGNOSE SVC	

Sorted according to macro names:

SVC number decimal	SVC number hexa.	Macro name (not to be used in system exits)	Comments (table sorted acc. to macro names)
222	DE	$AUDSW	
250	FA	$BOWT	
199	C7	$DOER	
39	27	$EOUT	cf. notes below
213	D5	$EXCP	
233	E9	$FNAT	
234	EA	$FNDT	
245	F5	$MASK	
235	EB	$PEND	
216	D8	$PGFR	
216	D8	$PGFX	
241	F1	$PNDT	
238	EE	$PNUP	
118	76	$SPORQ (RSO)	
108	6C	$SPRQ	
246	F6	$UNMASK	
236	EC	$UNPN	
214	D6	$WAIT	
215	D7	$XCPW	
125	7D	(DCAM calls)	
131	83	ARCHIVE	
36	24	BCAM-SVC	
92	5C	BKPT	
180	B4	BTAM (24-bit)	
157	9D	CATAL (24-bit)	
133	85	CATJV (24-bit)	
124	7C	CHKEI	
121	79	CHKSI	Serialization
54	36	CLCOM	
162	A2	CLOSE (24-bit)	
88	58	CMD (24-bit)	
145	91	CMD (31-bit)	
128	80	CONTXT	
16	10	CRYPT	
57	39	CSSCN	
75	4B	CSTAT (24-bit)	
98	62	DBSVC	
133	85	DCLJV (24-bit)	
187	BB	DELFEI	Eventing
121	79	DEQAR	Serialization
255	FF	DIAGNOSE SVC	
123	7B	DISCO	Contingency
124	7C	DISEI	

SVC number decimal	SVC number hexa.	Macro name (not to be used in system exits)	Comments (table sorted acc. to macro names)
122	7A	DISMP (24-bit)	
121	79	DISSI	Serialization
126	7E	DIV	
187	BB	DPOFEI	Eventing
187	BB	DSOFEI	Eventing
174	AE	ELIM (24-bit)	
123	7B	ENACO	Contingency
124	7C	ENAEI	
122	7A	ENAMP (24-bit)	
121	79	ENASI	Serialization
121	79	ENQAR	Serialization
133	85	ERAJV (24-bit)	
158	9E	ERASE (24-bit)	
60	3C	ERFLG	
128	80	EXIT	
178	B2	EXRTN (24-bit)	
94	5E	FASTACC	
94	5E	FASTPAM	
143	8F	FDDRL	
147	93	FEOV (24-bit)	
159	9F	FILE (24-bit)	
160	A0	FSTAT (24-bit)	
189	BD	FT-BS2000	
24	18	GEPRT (24-bit)	
151	97	GET (24-bit)	
164	A4	GETFL (24-bit)	
133	85	GETJV (24-bit)	
175	AF	GETKY (24-bit)	
176	B0	GETR (24-bit)	
163	A3	GOTO (24-bit)	
85	55	IC02000	
166	A6	IMPORT (24-bit)	
113	71	INITMB	
173	AD	INSRT (24-bit)	
141	8D	IOCFCOPY	
11	B	IOTRACE	
15	F	IPPMBASE	
109	6D	ITABL (24-bit)	
81	51	JSATTCH (24-bit)	
81	51	JSDETCH (24-bit)	
81	51	JSFXCPT (24-bit)	
81	51	JSINF (24-bit)	
81	51	JSINFO (24-bit)	
81	51	JSRUNDB (24-bit)	
134	86	KDCEND	

SVC number decimal	SVC number hexa.	Macro name (not to be used in system exits)	Comments (table sorted acc. to macro names)
178	B2	LBRET (24-bit)	
123	7B	LEVCO	Contingency
110	6E	LINK (24-bit)	
142	8E	LOADPM	
59	3B	MONTB	
93	5D	MSG (24-bit)	
38	26	MSG (31-bit)	
6	6	Meß-SVC	
10	A	NCAN	
14	E	NKDINF	
116	74	NLMSERVE	
12	C	NPCHK	
3	3	NPCSVC	
133	85	ONEVT (24-bit)	
50	32	OPCOM	
156	9C	OPEN (24-bit)	
47	2F	PLAM	
108	6C	PNCH	
184	B8	POSIX	
124	7C	POSSIG	
107	6B	PRNT	
152	98	PUT (24-bit)	
153	99	PUTX (24-bit)	
39	27	RDATA	cf. notes below
66	42	RDATA	
39	27	RDCRD	cf. notes below
62	3E	RDCRD	
160	A0	RDTFT (24-bit)	
53	35	RELBF	
161	A1	RELEASE (24-bit)	
74	4A	RELM (24-bit)	
122	7A	RELMP (24-bit)	
73	49	REQM (24-bit)	
122	7A	REQMP (24-bit)	
37	25	REQSPF	
123	7B	RETCO	Contingency
52	34	REVNT	
187	BB	RPOFEI	Eventing
187	BB	RSOFEI	Eventing
41	29	SAT	
29	1D	SCA	
86	56	SETBF	
133	85	SETJV (24-bit)	
154	9A	SETL (24-bit)	
51	33	SEVNT	

SVC number decimal	SVC number hexa.	Macro name (not to be used in system exits)	Comments (table sorted acc. to macro names)
114	72	SFDLOAD	
139	8B	SIR	
83	53	SM2	Usable only by SM2
124	7C	SOLSIG	
132	84	SPOOLFREE	
8	8	SRMASVC	
185	B9	SRMUINF	
133	85	STAJV (24-bit)	
133	85	STAM (24-bit)	
155	9B	STORE (24-bit)	
128	80	STXIT	
110	6E	SULNK	
123	7B	SUSPEND	Contingency
111	6F	TABLE	
9	9	TERM	Use $TERM
22	16	TERMD	
63	3F	TERMJ	
63	3F	TERMJD	
106	6A	UNLOD	
165	A5	VERIF (24-bit)	
13	D	VMMUSER	
112	70	VOLIN	
48	30	WARTOPT	
5	5	WRCPT (31-bit)	
39	27	WRLST	cf. notes below
69	45	WRLST	
39	27	WROUT	cf. notes below
67	43	WROUT	
39	27	WRTOT	cf. notes below
61	3D	WRTOT	
39	27	WRTRD	cf. notes below
68	44	WRTRD	
44	2C	YTCMAC	

6.2.3 Notes

SYSFILE environment

The RDATA and WRTRD macros generate wait states and, when processed, follow corresponding rules for using process levels and call hierarchies due to the link with break/escape processing (K2 key). Thus, correct execution of these functions can only be ensured in the case of a call from the TU state.

Like RDTAT/WRTRD, the WROUT macro requires an initialized SYSFILE environment.

DCAM calls

The DCAM calls cannot be used in the TPR state and are thus not applicable in system exits, because their execution causes a modification of process levels and involves the use of TPR contingencies. For this reason, the SVC itself must be called from a process level < 156 (=TU state).

ISAM SVCs

As a rule, the ISAM SVCs 32 and 168 can be used in system exits. However, take note of the following with regard to the SVC 168 in SHARUPD mode:

Most ISAM macros result in a release of external ISAM locks held by the job, so that only one lock is ever held and deadlock-free processing is ensured. Thus, a lock set by an application program may be reset by an ISAM call in a system exit without the application program noticing it.

Example

Application program:	GET	FCB1,AREA1,LOCK
System exit:	GET	FCB2,AREA2,LOCK
Application program:	PUTX	FCB1,AREA1

The PUTX macro is correctly executed without a system exit; with a system exit the lock is released and unpredictable results may occur.

For this reason, ISAM macros capable of releasing a lock should only be called in system exits during shared update processing when it is ensured that no application program is loaded. Even in this case, you must make sure that there is no undesired interaction with system files.

24-bit ISAM SVCs should no longer be used in system exits. Their use in an XS environment causes an error immediately, since system exits run in a TPR environment, i.e. in 31-bit mode.

JMS/JV SVCs

The JMS/JV SVCs are in a TPR environment, which means that they are executable in system exits. However, they do not always return meaningful results (e.g. the JINF macro does not if there is no task environment).

DSSM/SULNK

Only the SULNK interface is usable if the system exits have been loaded into privileged memory by DSSM or SULNK. If the system exits are loaded into nonprivileged memory, the SVCs VSVI, PBUNLD, ITABL, TABLE and LINK can be used. The $LOAD macro cannot be used.

Related publications

Please apply to your local office for ordering the manuals.

[1] **DCAM** (BS2000/OSD, TRANSDATA)
Macros
User Guide

Target group
Programmers of DCAM ASSEMBLER programs
Contents
– Special techniques when using DCAM macros
– DCAM macros, arranged according to functions
– Catalog of all DCAM macros

[2] **DCAM** (BS2000/OSD, TRANSDATA)
Program Interfaces
Reference Manual

Target group
– Managers
– Application planners
– Programmers
– System and network administrators
Contents
Description of the Data Communication Access Method DCAM

[3] **BS2000/OSD-BC**
Diagnostics Handbook
User Guide

Target group
The manual addresses system programmers, systems support and software maintenance.
Contents
The manual describes tools for identifying, logging and analyzing program execution data. It deals with dump generators (CDUMP, SNAP, SLED), dump analyzers (DAMP,NDMDAMP), logging tools (SERSLOG, AUDIT), the TRACE-MANAGER, and the log evaluator ELFE.

[4] **BS2000/OSD-BC**
Utility Routines
User Guide

Target group
The manual addresses both nonprivileged users and systems support.
Contents
The manual describes the utilities delivered with the BS2000 basic configuration BS2000/OSD-BC.

[5] **Distributed Print Services** (BS2000/OSD)
Printing in Computer Networks
User Guide

Target group
This manual is intended for nonprivileged users, device administrators and systems support of BS2000/OSD.
Contents
The manual provides descriptions of the principles, use and administration of Distributed Print Services for each of these user groups. Possible uses of Distributed Print Services are illustrated by examples.

[6] **BS2000/OSD-BC**
DMS Macros
User Guide

Target group
The manual addresses assembly language programmers.
Contents
The manual describes the DMS macro interface for the BS2000/OSD basic configuration. There is a brief description of the access method-specific features relevant to programming, followed by a description of the macros in alphabetical order.

[7] **BS2000/OSD-BC**
Introductory Guide to Systems Support
User Guide

Target group
This manual is addressed to BS2000/OSD systems support staff and operators.
Contents
The manual covers the following topics relating to the management and monitoring of the BS2000/OSD basic configuration: system initialization, parameter service, job and task control, memory/device/system time/user/file/pubset management, assignment of privileges, accounting and operator functions.

[8] **BS2000/OSD-BC**
Commands (several volumes)
User Guides

Target group
The manual addresses both nonprivileged BS2000/OSD users and systems support.
Contents
This manual contains BS2000/OSD commands (basic configuration and selected products) with the functionality for all privileges. The introduction provides information on cmd input.

[9] **BS2000/OSD-BC**
Commands, Volume 6, Output in S Variables and SDF-P-BASYS
User Guide

Target group
This manual is addressed to programmers and users who write procedures.
Contents
Volume 6 contains tables of all S variables that are supplied with values by the SHOW commands in conjunction with structured output. Further chapters deal with:
- introduction to working with S variables
- SDF-P-BASYS

[10] **BS2000/OSD-BC**
Executive Macros
User Guide

Target group
The manual addresses all BS2000/OSD assembly language programmers.
Contents
The manual contains a summary of all Executive macros, detailed descriptions of each macro with notes and examples, including job variable macros, and a comprehensive general training section.

[11] **SDF-A** (BS2000/OSD)
User Guide

Target group
This manual is intended for experienced BS2000 users and system administration staff.
Contents
It describes how to process syntax files and explains the SDF-A functions on the basis of examples. The SDF-A statements are listed in alphabetical order.
The manual also includes a description of the SDF-SIM utility routine.
SDF-A V4.1A can be used as of BS2000/OSD-BC V1.0.

[12] **SDF** (BS2000/OSD)
Introductory Guide to the SDF Dialog Interface
User Guide

Target group
BS2000/OSD users
Contents
This manual describes the interactive input of commands and statements in SDF format. A Getting Started chapter with easy-to-understand examples and further comprehensive examples facilitates use of SDF. SDF syntax files are discussed.

[13] **SECOS** (BS2000/OSD
Security Control System
User Guide

Target group
– BS2000 system administrators
– BS2000 users working with extended access protection for files
Contents
Capabilities and application of the functional units:
– SRPM (System Resources and Privileges Management)
– SRPMSSO (Single Sign On)
– GUARDS (Generally Usable Access Control Administration System)
– GUARDDEF (Default Protection)
– GUARDCOO (Co-owner Protection)
– SAT (Security Audit Trail).

[14] **SPOOL** (BS2000/OSD)
User Guide

Target group
This manual is intended for nonprivileged users, Spool & Print administrators, RSO device administrators and systems support staff.
Contents
The manual describes the operation of SPOOL.

[15] **BS2000/OSD-BC**
System Installation
User Guide

Target group
This manual is intended for BS2000/OSD system administration.
Contents
The manual describes the generation of the hardware configuration with UGEN and the following installation services: disk organization with MPVS, the installation of volumes using the SIR utility routine, and the IOCFCOPY subsystem.

[16] **BS2000/OSD-BC**
System Messages (several volumes)
User Guide

Target group
This manual is addressed to systems support staff, operators and users.
Contents
Chapter 1 deals with message processing in BS2000/OSD. Chapter 2 contains the system messages for the basic configuration of the BS2000/OSD operating system. The messages are arranged in alphabetical order by message class.

[17] **DSSM/SSCM**
Subsystem Management in BS2000/OSD
User Guide

Target group
This manual addresses systems support staff and software consultants of BS2000/OSD.
Contents
The following are described: BS2000/OSD subsystem concept, dynamic subsystem management (DSSM), subsystem catalog management (SSCM) and the associated commands and statements.

[18] **BS2000/OSD - Technische Beschreibung**
(Technical Description, currently available in German only)

Target group
BS2000 users with an interest in the technical background of their systems (software engineers, systems analysts, computer center managers, system administrators) Computer scientists interested in studying a concrete example of a general-purpose operating system.

Index

Contents

BS2000/OSD-BC V5.0
System Exits

User Guide

Target group
The manual addresses systems support.
Contents
The manual contains an introduction to the system exits with a description of the base mechanism. The main section comprises descriptions of all system exits, e.g. exits for the BS2000/OSD basic configuration, SPOOL, DCAM, SDF and PLAM.

Edition: March 2002

File: sysexit.pdf

This manual was produced by
cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Fujitsu Siemens computers GmbH
User Documentation
81730 Munich
Germany

Fax: (++49) 700 / 372 00000

e-mail: manuals@fujitsu-siemens.com
http://manuals.fujitsu-siemens.com

Comments
Suggestions
Corrections

Submitted by

Comments on BS2000/OSD-BC V5.0
System Exits

Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format *...@ts.fujitsu.com*.

The Internet pages of Fujitsu Technology Solutions are available at *http://ts.fujitsu.com/*...
and the user documentation at *http://manuals.ts.fujitsu.com*.

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form *...@ts.fujitsu.com*.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter *http://de.ts.fujitsu.com/*..., und unter *http://manuals.ts.fujitsu.com* finden Sie die Benutzerdokumentation.