

# Preface

The "User Commands (ISP Format)" manual contains descriptions of the commands available to non-privileged users of the BS2000 operating system. The special commands for the system administrator and the operator can be found in the manuals "System Administrator's Guide" [1] and "System Operator's Guide" [2].

In addition to an introduction to the operating system and an overview of all available commands arranged according to application areas, this manual includes detailed descriptions of the commands in alphabetical order, with formats, explanations and examples.

## What previous knowledge is required?

The user should have a basic knowledge of BS2000 such as is provided by the manual "Introductory Guide for System Users" [3].

## Literature references

All literature references in the text are given in abbreviated form. Complete titles of every publication referred to can be found under "References". This is followed by a brief note on how to order manuals.



As of BS2000 V10, the SDF user interface is no longer unloadable and is therefore accessible at all times. Since SDF is to replace, on a medium-term basis, the (ISP) user interface which has been in use thus far, some of the changes made in the existing BS2000 Version 10 are no longer implemented in ISP. All changes made to the operating system after BS2000 V10 will only be available through the SDF user interface. This edition of the manual thus represents in all likelihood the latest edition for user interface ISP.

### Changes in BS2000 V10.0A since the last version BS2000 V9.5A

#### **SDF**

In V10.0A SDF is accessible in the system at all times and can no longer be unloaded.

#### **Data/information security**

New basic access control mechanisms (BACL) for data access are provided as part of the BS2000 basic configuration. The software product SECOS furthermore adds the following functions:

- stricter access control with improved identification and authentication of users (chipcard, password, lifetime, etc.)
- new file access control options
- introduction of user groups
- mechanisms for extended management of access rights and decentralization of system administration (assignment of system privileges)
- auditing and logging of events relevant to security (access to protected files, input of passwords, etc.)

#### **PVSID extension and shared pubsets**

Extension of the PVSID (pubset identification) and the CATID (catalog identification) to up to 4 characters has considerably increased the number of Public Volume Sets (PVS) that can be operated concurrently.

MSCF V10.0 allows you to access files from more than one computer. Up to 16 systems are contained in one integrated computer network. One of the computers in the network is declared the pubset owner; it then handles the file access management functions for all systems.

#### **Extension of TSNs**

Alphanumeric TSNs can also be permitted for user tasks at system generation time, thus considerably increasing the number of jobs that the job management facility can handle simultaneously.

**Elimination of the PAM key**

BS2000 V10.0 also formats and operates disk storages without key fields (FBA format), which results in improved utilization of storage capacity. The non-key disk format allows the new non-key access methods NK-SAM and NK-UPAM to be used. Non-key operation must be configured at pubset level, which means that all disks in a given pubset must be operated either with key fields or all of them must be operated without key fields. You can use secondary keys in non-key ISAM (NK-ISAM) under Version 10.0A. Thus, in addition to the primary key, a number of fields are defined as key fields and records are processed by means of several search criteria (keys).

**Periodic accounting**

The periodic accounting facility continually records accounting data. For long-running programs in particular, this ensures that the data required for virtually complete accounting is available, even if unexpected events occur. Up to 16 job classes can be defined for such periodic recording. In the case of tasks from a monitored job class and with a program loaded at the time in question, the program utilization data for those tasks is recorded continuously and stored in the system accounting file.

**Hardware support**

As of Version 10.0A, card readers and card punches are no longer supported. The software product ADAM is now required in order to operate them.

## Summary of new command operands

Command	Operand	Meaning
COPY	BLKCTRL	Determines format for target file
CREATE-ISAM-POOL	WRITE-IMMEDIATE	Saves updates immediately
EXECUTE	RUN-MODE	Determines DBL operating mode
	VPSPACE	Reserves vector storage space
	VPWAIT	Waits for vector storage space reservation
FSTATUS	ACL	File selection on the basis of ACL entry
	BASIC-ACL	File selection on the basis of BACL entry
	LASTPAGE	File selection on the basis of the number of PAM pages written
PRINT	LIBRARY-ELEMENT	Outputs library member
	FAMILY	Collects a number of jobs under one task
	TRUNC	Deletes file after error
	RESOURCE=LP65	New printer type
	CCPOS	Specifies control character position
	PAGECC	Control character [not] in the file
	DUSER	Specifies foreign user ID
	DACCOUNT	DUSER's account number
	DPASSW	DUSER's password
	SECTION	Delimits output area
	CHKPT	Sets checkpoints
	IMAGE	Specifies character set file
	CONTROL	Defines control character analysis
RESTART	TEMPJV	Tests temporary job variable
STATUS	TERMINAL	Displays DCAM name
STAM	SELECT	Selects MRSCAT entries

### The following are no longer supported:

- The MARGIN command
- Output to punch cards or magnetic tape by means of the PUNCH command.  
Exception: POOLER tape.

## Functions of the user commands

The user commands enable you to direct the **Control System** of the BS2000 operating system.

These commands are addressed to the four components of the Control System:

- Executive,
- Data Communication Methods (DCM),
- Data Management System (DMS) and
- System services

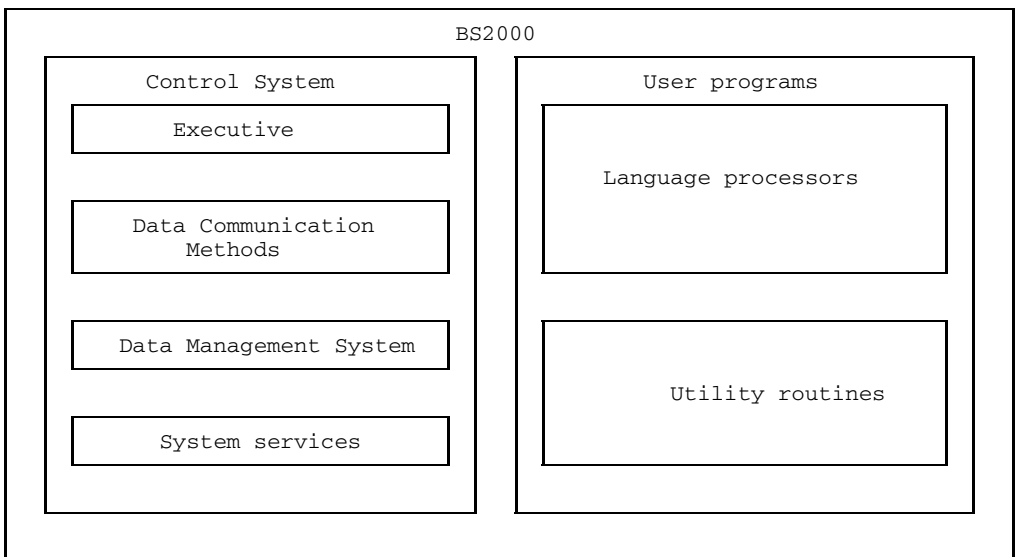


Fig. P-1 BS2000 components

The **Executive** contains the operating system's central control routines and performs the following tasks:

- control of the execution of all jobs, e.g. all interactive, batch and spooling jobs;
- real and virtual memory management;
- formal analysis of commands;
- execution of spooling operations;
- console input and output;
- system accounting.

BS2000 commands which are related to jobs, influence job and program execution, and concern the console or system accounting are thus directed to the Executive.

The **Data Communication Methods** (TRANSDATA DCM) perform the following tasks:

- data communication between program and terminals or other programs;
- management of the requisite resources.

BS2000 commands that control the operation of data display terminals, data printers, and batch terminals are directed to the DCM.

The **Data Management System** (DMS) contains routines designed to perform the following functions:

- file management, such as the cataloging, storage, retrieval and deletion of files;
- support for file access methods;
- input and output to the peripheral devices (except consoles and terminals).

BS2000 commands for purposes of file, volume, and device handling refer to the DMS.

The **system services** include additional Control System functions such as the Dynamic Linking Loader (DLL).

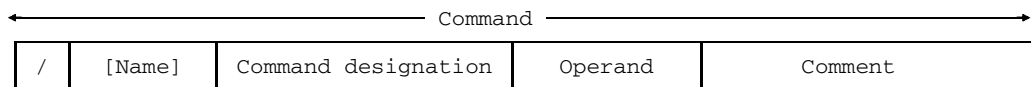
Besides the user commands there are

- commands for the system administrator (see the "System Administrator's Guide" [1]) and
- commands for the operator. These commands are entered from a console (see the "System Operator's Guide" [2]).

## Command formats

The BS2000 commands all have the same format. They all start with a slash, which identifies them to the operating system and the user as commands. In interactive mode, the slash is provided by the operating system, while in batch mode you must specify it in the first column of each command line.

The slash is followed by entries which can be divided into four parts:



### Name

**Purpose:** The name specifies a symbolic address which can be used as a branch destination in other commands. You may choose whether or not to assign a name to the command. The representation of the name is generally omitted from the command formats in the chapter "User Commands", starting on page 39, except for those cases where a name is of particular importance.

**Format:** The name may start with any number of blanks (X'40'). A period identifies the start of the name; this is followed by a letter or one of the characters @, #, or \$, which may in turn be followed by up to 7 characters (alphanumeric or @, #, \$). The name may end in a blank.

#### *Example*

```
/.NAME1 .....
/      .XY .....
/.A1234567 .....
```

### Command designation

**Purpose:** The operation to be executed is specified by the command designation or its abbreviated form. This entry must therefore appear in every command.

**Format:** The command designation may start with any number of blanks (X'40'). If a name is specified, a blank must be inserted in front of the command designation. If no name is specified, the name of the command or its abbreviation may immediately follow the slash.

### *Example*

```
/ERASE  
/.NAME FSTATUS  
/.X STEP
```

### **Operand**

**Purpose:** Operands define how a command is to be executed in each case.

**Format:** The first operand must be separated from the command designation by at least one blank (X'40'). Successive operands must be separated by commas. Any number of blanks may be used before or after commas. The same applies to equal signs in keyword operands.

**Positional operands** are defined by their position in the command and must therefore be specified in strict accordance with a predetermined sequence. If a positional operand is omitted, the separating comma for the next positional operand must nevertheless still be specified in order to define the position of the following positional operand.

**Keyword operands** are defined by means of a keyword (KEYWORD=) and can therefore appear in any sequence.

Both types of operands may appear in a command.

### *Example*

```
/PRINT DATA } Positional  
/.NEWNAM CATALOG NAME1 } operands  
  
/FILE LINK=LINK,DEVICE=TAPE } Keyword  
/.COMPIL PARAM ERRFIL=YES } operands  
  
/ERASE FILE,DATA,LIST=YES } 2 positional,  
 } 1 keyword  
 } operand
```



## Comment

- Purpose:** Any desired information can be written in the form of a comment, since the system does not interpret this area. It is not essential to enter comments.
- Format:** The comment may start with one or more blanks (X'40'). The text itself must be enclosed in quotes (") and may contain any characters except quotes and ETX characters. It may be inserted immediately after operands, or also before or after a comma separating operands.

### *Example*

```
/FSTAT "**FILE STATUS FOR ALL FILES**"  
/.AIM EXECUTE PROG"TEST PROGRAM NO. 5"
```

## Continuation lines

A command may consist of one or more lines.

The continuation character is the hyphen "-". It indicates the presence of a continuation line. Only blanks are allowed between the continuation character and the end of the line. Each continuation line begins with a slash, which is either generated automatically by the system or - in the case of procedure files - must be entered by you as user. Depending on system generation (system parameter SSMCOPT), two different continuation mechanisms may occur:

- Continuation mechanism for BS2000 V7.1 (SSMCOPT=N) with a fixed area for the continuation character for command records in files.
- Continuation mechanism for BS2000 versions as of V7.5 (SSMCOPT=Y) with a variable area for the continuation character.

### Continuation mechanism for SSMCOPT=N

A command may have up to 13 continuation lines: however, the number of characters must not exceed 1016.

### *Exception*

Only 11 continuation lines are allowed for the following commands on

- on remote card reader: LOGON, LOGOFF, DATA, END, RJOB, RLOGOFF, RMSG, ROUT, RSTART, RSTATUS, RSTOP;
- on floppy disk: LOGON, LOGOFF, DATA, END;

The maximum input length allowed for a command and the area in which the continuation character may be located both depend on the input source, as shown in the table below:

Input source	Input length	Continuation char.
Terminal	Up to 1016 characters (also dependent on input buffer).	In the area from column 2 to the end of the input record.
Procedure file, ENTER file, floppy disk	Up to and including column 71	Column 72



Blanks following continuation characters are ignored. Procedure file commands with more than 13 continuation lines will trigger a SPIN-OFF (branch to next ABEND, ABORT, STEP or LOGOFF command).

### Example

```
Line 1: /file file name,-
Line 2: /link=dset10,"linkname"-
Line 3: /device=tape,fcctype=sam
```

```
} Terminal
```

```
Column
 72
```

```
Line 1: /FILE FILENAME,
Line 2: /LINK=DSET10,
Line 3: /FCCTYPE=SAM
```

```
- } Procedure file, ENTER
- } file, floppy disk
```

### Continuation mechanism for SSMCOPT=Y

A command may have any number of continuation lines; however, the total number of characters must not exceed 2036.

#### *Exception*

Only 13 continuation lines are allowed for the following commands:

- on remote card reader: LOGON, LOGOFF, DATA, END, RJOB, RLOGOFF, RLOGON, RMSG, ROUT, RSTART, RSTATUS, RSTOP
- on floppy disk: LOGON, LOGOFF, DATA, END

The maximum input length allowed for a command and the area in which the continuation character may be located both depend on the input source, as shown in the table below:

Input source	Input length	Continuation char.
Terminal	Max. 2036 characters.	In the area from column 2 to the end of the input record
Procedure file, ENTER file	Up to and including column 71	In the area from column 2 to 72
Floppy disk	Up to and including column 71	In the area from column 2 to 72 with the exception of the LOGON, LOGOFF, DATA, END commands (col. 72)

- ❗ Blanks following the continuation character are ignored. A command with more than 2036 characters is not logged and triggers a SPIN-OFF (branch to next ABEND, ABORT, STEP or LOGOFF command).
- ❗ If the REMARK command text contains hyphens in the area from column 2 to 71, the last hyphen is interpreted as a continuation character. Any subsequent command is ignored since it is considered to be a continuation of the REMARK command.

### Example

```
Line 1: /FILE FILENAME,-
Line 2: /LINK=DSET10,"linkname"-
Line 3: /DEVICE=TAPE,FCBTYPE=SAM
```

```
} Terminal,
} procedure file,
} ENTER file
```

```
Line 1: /LOGON
Line 2: /USERID,
Line 3: /M3954TRS
```

```
Column
72
- } LOGON, END, LOGOFF,
- } DATA commands on
} floppy disk
```

## Conventions used in command descriptions (metacharacters)

Specific characters (so-called metacharacters) are used and special conventions apply for the representation of the command format. They are explained in this table:

Format notation	Explanation	Example
UPPERCASE	Uppercase letters indicate constants which you must enter in precisely this form.	/SYSFILE SYSDTA=(SYSCMD) ..... You enter: /SYSFILE SYSDTA=(SYSCMD)
lowercase	Lowercase letters indicate variables for which you must substitute appropriate values when entering the command, i.e. their contents can vary from case to case.	/PRINT filename ..... You enter: /PRINT FILE, /PRINT XYZ, /PRINT A.B-1, etc.
{ }	Braces are used to indicate alternatives, i.e. an entry can or must be selected from the enclosed values.	{YES} {NO} ..... You enter: YES or NO
	A vertical line separates alternative entries.	NONE   password ..... You enter: NONE or, e.g., C'XXX'
[ ]	Square brackets indicate that the enclosed entries are optional, i.e. may be omitted. If, in the case of optional entries, the comma is inside the brackets, it must only be written when that operand is used and can be omitted before the first operand in a command or a sequence of operands. If, on the other hand, it is outside the brackets, it must always be written, even if no optional entries are made. (Parentheses must be entered.)	tsn [,DUMP] ..... You enter, for example: 1028 or 1028,DUMP

Format notation	Explanation	Example
—	Underscoring (underlining) indicates the default value, i.e. the value used by the system if you have not specified anything.	<pre> { <u>ISAM</u> } [ { SAM } ] ..... You enter: SAM or ISAM or nothing (=ISAM) </pre>
.....	Dots indicate repetition, i.e. the preceding syntactical unit may be repeated several times in succession.	<pre> (vsn,...) ..... You enter: (PVT003) or (PVT003,PVT456) or (XY00AB,XY0012,XY0005) etc. </pre>
_	This character indicates a blank (X'40')	<pre> STD_ ..... You enter: 'STD ' </pre>

Apart from the position of continuation characters, the command format remains the same, regardless of whether the command is entered in batch or interactive mode. However, not every command can be used in either mode.

## List of user commands

Command	Function	Available as macro	Additional software product
ABEND	Terminate job issuing command	CMD 1)	
ABORT	Abort procedure	-	
ADD-ISAM-POOL-LINK	Define ISAM pool link name	ADDPLNK	
APPLICATION	Enter DCAM application in CLT	-	
AUDIT	Activate audit mode	AUDIT 3)	
BCNTRL	Suppress operator messages	CMD 1)	
BREAK	Request system mode	-	
CALL	Initiate procedure file	CMD 1)	
CANCEL	Cancel job	CMD 1)	
CANCEL-FILE-TRANSFER	Terminate file transfer	CMD 1)	FT
CATALOG	Process catalog entry	CATAL 4)	
CATJV	Create job variable	CATJV 2)	JV
CHANGE	Change TFT entry	CHNGE 4)	
CONNECTION	Enter virtual connection in CLT	-	
COPY	Copy file	COPY 4)	
CREATE-ISAM-POOL	Create or join ISAM pool	CREPOOL	
DATA	Open DATA file	-	
DCLJV	Define job variable link name	DCLJV 2)	JV

Command	Function	Available as macro	Additional software product
DELON	Delete ON command	DONEVT	JV
DELETE-ISAM-POOL	Delete or sever link to ISAM pool	DELPPOOL	
DO	Initiate procedure file	CMD 1)	
DROP	Cancel HOLD status	-	
END	Close DATA file	-	
ENDON	Terminate ON statement sequence	-	JV
ENDP	Terminate procedure file	-	
ENTER	Initiate ENTER job	ENTER 2)	
EOF	Indicate file end for SYSDTA	-	
ERAJV	Erase job variable	ERAJV 2)	JV
ERASE	Erase file	ERASE 4)	
ESCAPE	Interrupt procedure	-	
EXECUTE	Load and start program	-	
FILE	Define file attributes	FILE 4)	
FSTATUS	Request catalog information	FSTAT 4)	
GETJV	Output job variable value	GETJV 2)	JV
GETUS	Read user switches	CMD 1) GETUS 3)	
HELP	Output message text	CMD 1)	
HOLD	Lock TFT entry	-	
IMPORT	Create catalog entry for files on private disks	IMPORT 4)	

## User commands

Command	Function	Available as macro	Additional software product
INTR	Start interrupt routine for a loaded program	STXIT 3)	
LOAD	Load program	-	
LOGOFF	Terminate job	LOGOFF 2)	
LOGON	Initiate job	-	
MODIFY-JOB	Modify job attribute		
MODIFY-JV-CONDITIONALLY	Check and set job variable		JV
MODIFY-MSG-ATTRIBUTES	Define language for message output, define scope	MSGSMOD	
MRSSTA	Output multiprocessor system status	MRSSTA 2)	MSCF
MSGCONTROL	Add/delete (task-specific) message files	MSGSMOD	
ON	Initiate conditional execution of a command sequence	ONEVT 3)	JV
OPTION	Specify job logging	CMD 1)	
PARAMETER	Control language processors	CMD 1)	
PASSWORD	Specify password	CMD 1)	
PAUSE	Inform operator and wait	CMD 1)	
PRINT	Print file	CMD 1) PRINT 3)	
PRIORITY	Alter job or run priority	CMD 1)	
PROCEDURE	Specify procedure file attributes	- -	



Command	Function	Available as macro	Additional software product
PSWORD	Protect user ID by means of password	CMD 1)	
PUNCH	Output file	CMD 1) PNCH 3)	
RDFTFT	Information from TFT and TST	RDFTFT 4)	
RELEASE	Delete TFT entry	REL 4)	
REMARK	Insert remark into command file	-	
REMOVE-ISAM-POOL-LINK	Remove ISAM pool link name	REMLNPK	
RESTART	Restart program at checkpoint	WRCPT 3)	
RESUME	Change from system mode to program mode	-	
RFAEND	Clear down RFA connection	-	RFA
RFASTART	Set up RFA connection	-	RFA
RJOB	Name batch job	-	
RLOGOFF	Terminate batch terminal input	-	
RLOGON	Initiate batch terminal input	-	
RMSG	Issue message from batch terminal	-	
ROUT	Request job output	-	
RSTART	Activate batch terminal	-	
RSTATUS	Request status of remote batch job	-	
RSTOP	Deactivate batch terminal	-	

## User commands

Command	Function	Available as macro	Additional software product
RTI	Return to interrupted procedure	-	
SECURE	Request resources	-	
SETJV	Set job variable	SETJV 2)	JV
SETSW	Read or alter job switches	CMD 1) SETSW 3)	
SETUS	Alter user switches	CMD 1) SETUS 3)	
SET-SS-OPTION	Release reserved address space for subsystem		
SHOW-CJC-STATUS	Output information on CJC function (conditional job control)		JV
SHOW-DEVICE-CONFIGURATION	Output information on system configuration	DSTATUS	
SHOW-DEVICE-STATUS	Output information on volumes (occupancy and volume monitoring data)	DSTATUS	
SHOW-DISK-DEFAULTS	Output default values for DISK parameters	DSTATUS	
SHOW-DISK-STATUS	Output disk occupancy and DISK parameter settings	DSTATUS	
SHOW-FILE	Output a file or library member to the screen		
SHOW-FILE-TRANSFER	Output information on FT jobs		
SHOW-JOB-CLASS	Output job class definitions		

Command	Function	Available as macro	Additional software product
SHOW-JV-LINK	Output link names for job variable		JV
SHOW-MOUNT-PARAMETER	Output mount settings for data volumes	DSTATUS	
SHOW-MSG-DEFAULTS	Supply information about the number, names, language and scope of message files	MSGSHOW	
SHOW-RESOURCE-ALLOCATION	Output settings and open operator actions for a task	DSTATUS	
SHOW-SPOOL-CHARACTER-SETS	Output character sets defined with the system administrator command ADD-SPOOL-CHARACTER-SET		
SHOW-SPOOL-DEVICES	Output devices defined with the system administrator command ADD-SPOOL-DEVICE		
SHOW-SPOOL-FORMS	Output forms defined with the system administrator command ADD-SPOOL-FORM		
SHOW-SPOOL-PARAMETER	Output current parameter listing		
SHOW-TAPE-STATUS	Output information on specified tapes and devices	DSTATUS	
SHOW-USER-ATTRIBUTES	Output data of entry in JOIN file	AREC	
SKIP	Branch depending on job switch	-	
SKIPJV	Conditional branch to command sequence	-	JV
SKIPUS	Branch depending on user switch	-	

Command	Function	Available as macro	Additional software product
SPARAM	Compress SPOOL output	-	
STAJV	Output status of job variables	STAJV 2)	JV
STAM	Read MRSCAT entries	STAM 2)	MSCF
STATUS	Request status information on jobs	CMD 1)	
STEP	Set procedure section	-	
SYSFILE	Assign system files	SYSFL 2)	
SYSTATUS	Output information on system files	SYSTA 2)	
TCHNG	Change terminal characteristics	TCHNG 3)	
TRANSFER-FILE	Transfer file	CMD 1)	FT
TYPE	Inform operator	CMD 1) TYPIO 3)	
VERIFY	Reconstruct file	VERIF 4)	
WAIT	Initiate conditional wait state	-	JV
WHEN	Enter conditional wait state	-	
WRITE-ACCOUNTING-RECORD	Write user accounting record	AREC	

- 1)  
The command may be called via the CMD macro.
- 2)  
The command may be called via CMD or as an MCLP macro (see the "Executive Macros" manual [5]).
- 3)  
The function of the command is similar to that of the macro or complements it (see the "Executive Macros" manual [5]).
- 4)  
The command may be called via CMD or as an MCLP macro (see the "DMS" manuals [8, 9]).

## List of user commands and brief description according to application areas

This chapter groups the commands according to application areas. The individual commands, which may appear in more than one list, are accompanied by brief descriptions characterizing their functions.

This overview is intended to help the user find the commands more quickly.

## Commands for controlling job execution

### Job initiation and termination

Command	Brief description of function
ABEND	Force abnormal termination of the job
CANCEL	Terminate or cancel a job
ENTER	Initiate a new batch job
LOGOFF	Terminate a batch or interactive job, and initiate a SPOOL job for the system output files (SYSLSST, SYSOUT, SYSOPT) if required
LOGON	Start a batch or interactive job; you must identify yourself to the system
MODIFY-JOB	Modify the job attributes of a batch job
PRINT	Output files to a printer, batch terminal or magnetic tape by creating a SPOOL job
PSWORD	Define, change or delete the password protecting your user ID
PUNCH	Output files to magnetic tape/floppy disk by generating a SPOOL job
SPARAM	Compress subsequent outputs of the PRINT command

## Priorities and job accounting

Command	Brief description of function
LOGON	Start a job and indicate to BS2000 the user ID and account number; may also be used to define job priorities
MODIFY-JOB	Modify the attributes of a batch job
PRIORITY	Change job priority
WRITE-ACCOUNTING-RECORD	Write a user accounting record

## Job logging

Command	Brief description of function
AUDIT	Log the audit mode for function state P1; branch instruction addresses are entered in the AUDIT table
LOGON	Start a job; may also be used to specify job logging
OPTION	Change the type of logging during job execution and specify whether to output a dump in the event of error
REMARK	Write comments to SYSOUT
STATUS	Interactively request information about the status of current jobs

**System messages and dialog with the operator**

Command	Brief description of function
BCNTRL	Suppress the output of operator messages
HELP	Display explanatory text on the screen for system messages
LOGON	Start a job; may also be used to specify the logging mode for the job and the system message format
OPTION	Define the logging mode for system messages of the job and whether console messages are to be logged
PAUSE	Display a message on the operator's console; after confirmation by the operator, the job is continued
RMSG	Output a message to the operator in batch mode
SHOW-MSG-DEFAULTS	Interrogate the number, names, language and scope of message files
TYPE	Output a message to the operator

**Procedure files**

Command	Brief description of function
ABORT	Interrupt a procedure; SYSCMD returns to the primary command input
DO CALL	Execute a command sequence which is stored in a procedure file
ENDP	End the procedure (last command in a procedure file)
ESCAPE	Interrupt a procedure (ESCAPE mode) in order to enter commands via the terminal
PROCEDURE	Define the start of a procedure file. A procedure file contains commands, and possibly data, which can be executed as part of a batch or interactive task
RTI	Exit ESCAPE mode and return to the interrupted procedure
STEP	Mark the position in a procedure from where the procedure is to continue after command input error



### Job switches and branches during command execution

Command	Brief description of function
SETSW	Interrogate, set, reset and invert the 32 job switches possessed by each job
SKIP	Branch to specific positions within a procedure (depending on the setting of job switches)
STEP	Deactivate job switches 16 through 31

### User switches and branches during command execution

Command	Brief description of function
GETUS	Display the setting of user switches
SETUS	Alter the setting of user switches allocated to each user ID (can be used to set, reset, and invert switches)
SKIPUS	Branch to specific positions within a procedure, depending on the setting of user switches which must be specified. Unconditional branches are also possible
WHEN	Interrupt a job, depending on the setting of specified user switches

## Commands for file processing

### File creation and file processing

Command	Brief description of function
ADD-ISAM-POOL-LINK	Enter an ISAM pool link name in the table of pools
CREATE-ISAM-POOL	Create an ISAM pool or set up a link to an ISAM pool
DELETE-ISAM-POOL	Delete an ISAM pool or sever a link to an ISAM pool
DATA	Catalog and open a file for spoolin from floppy disk. Subsequent records are written to this file.
END	Close a file opened by a DATA command
REMOVE-ISAM-POOL-LINK	Remove an ISAM pool link name from the table of pools

### File copying, entry and output

Command	Brief description of function
COPY	Copy a file that is in standard block format to disk or tape
FILE	Enter file attributes (using the operand STATE=FOREIGN) of an existing but as yet uncataloged file (on tapes or private disks) into the catalog
IMPORT	Generate catalog entries for existing files on private disks
PRINT	Output files to printers, batch terminals or tapes
PUNCH	Output files to magnetic tape/floppy disk or POOLER tape
SHOW-FILE	Display a file or library member on the screen
SYSFILE	Specify the print format for system output files SYSLST and SYSOPT

## Catalog processing

Command	Brief description of function
CATALOG	Generate or change a file's catalog entry
ERASE	Delete the user's own files and their catalog entries
FILE	Generate and, where necessary, change a catalog entry, i.e. the entries pertaining to volume, device and space allocation associated with the file
FSTATUS	Interrogate file attributes

## File protection

Command	Brief description of function
CATALOG	Used to define the following file attributes: file protection passwords; shareability; read, write, or execute access authorization; file retention period
FILE	Specify the retention period for a file
PASSWORD	Enter the file protection passwords (read/write) into the password list for the job
SECURE- RESOURCE- ALLOCATION	Reserve a file during job execution
SECURE	
VERIFY	Make a file which was abnormally closed available again

### File deletion

Command	Brief description of function
ERASE	Selectively delete your own files (both the data and the catalog entries) and release the storage space
PRINT	Delete files automatically after output to printer, batch terminal or tape
PUNCH	Write files to floppy disk, tape or pooler tape and then delete them automatically

### Processing of file generation groups

Command	Brief description of function
CATALOG	Used to generate or change the catalog entry for a file generation group, especially to define file protection for the FGG
COPY	Copy file generation groups or individual file generations
ERASE	Delete file generation groups and individual file generations
FILE	Define attributes of file generations and generate any catalog entry required
FSTATUS	Interrogate file generation groups or individual file generations
PRINT	Output individual file generations to printer or tape
PUNCH	Output individual file generations to magnetic tape/floppy disk or POOLER tape
SECURE- RESOURCE- ALLOCATION	Lock a file generation group against unauthorized access
SECURE	

## Commands for device and volume reservation

Command	Brief description of function
CATALOG	Specify the device and volume requirements for file generation groups
CHANGE	Change a TFT entry (file link name)
DROP	Cancel the hold status on RELEASE
FILE	Specify the device and volume requests for a file. The MOUNT operand controls the number of requests
HOLD	Delay the effect of the RELEASE command until a DROP command is processed
RELEASE	Delete the file link name in the task file table (TFT) and release associated private volumes, and possibly also devices
RDTFT	Obtain information on devices and volumes which are linked to a certain file
SECURE- RESOURCE- ALLOCATION  SECURE	Reserve resources (devices for private volumes, private volumes, and files). The reservation may be either exclusive or shareable. When entered without operands, the command releases all resources reserved for the job
SET-SS- OPTIONS	Release address space (class 5 memory) reserved for subsystems
WHEN	Interrupt a job, depending on the setting of user switches. Devices assigned by the job are released

The commands LOGOFF, ABEND and CANCEL terminate a job and release its resources.

## Commands for program control

### Language processing

Command	Brief description of processing
PARAMETER	Provide precise specifications for the assembler and the COBOL, FORTRAN, ALGOL, PL/I and RPG2 compilers, and enter them in the parameter list of the job
STEP	Reset the values in the parameter list of the job to the default values for the PARAMETER command
SYSFILE	Assign the file with the source program to the SYSDTA system file

### Program loading and starting

Command	Brief description of function
EXECUTE	Load a program (i.e. object module, load module or LLM) into memory and start it
LOAD	Load a program (i.e. object module, load module or LLM) into memory without starting it
RESUME	Start or continue a loaded program
SYSFILE	Specify an object module file for the Dynamic Linking Loader (DLL) by means of the TASKLIB operand

### Program interrupts and restarts

Command	Brief description of function
BREAK	Interrupt input from SYSDTA if SYSDTA and SYSCMD are combined
EOF	Generate an EOF condition for system file SYSDTA
INTR	Transfer control to the STXIT routine of the loaded program
RESUME	Return from the command mode to the user program (see the "Interactive Debugging Aid" manual [4])
RESTART	Restart a program for which a checkpoint was set

## Commands for interrogating current values

Command	Brief description of function
FSTATUS	Interrogate file attributes
GETUS	Display the setting of the user switches
RDTFT	Display status information on the files and tape devices currently in use, as well as on the linked TFT and TST
RSTATUS	Ascertain the status of one's own batch jobs
SETSW	Display the current setting of job switches (the command must be entered without operands)
SHOW-DEVICE-CONFIGURATION	Interrogate the system configuration
SHOW-DEVICE-STATUS	Interrogate the occupancy of physically online volumes
SHOW-DISK-DEFAULTS	Display the default values for the DISK parameters
SHOW-DISK-STATUS	Interrogate occupancy, DISK parameters and volume monitoring for the specified disks
SHOW-FILE-TRANSFER	Display information on FT jobs
SHOW-JOB-CLASS	Interrogate the job class definitions
SHOW-MOUNT-PARAMETER	Interrogate the default values for mounting and dismounting volumes
SHOW-MSG-DEFAULTS	Ascertain the number, names, language and scope of the message files
SHOW-RESOURCE-ALLOCATION	Output the device assignment for a task running under your own user ID

## Interrogation

---

Command	Brief description of function
SHOW-SPOOL-CHARACTER-SETS	List character sets defined by the DMS command ADD-SPOOL-CHARACTER-SET
SHOW-SPOOL-DEVICES	List devices defined with the DMS command ADD-SPOOL-DEVICE
SHOW-SPOOL-FORMS	List the forms defined with the DMS command ADD-SPOOL-FORM
SHOW-SPOOL-PARAMETERS	Display the current SPOOL parameter list
SHOW-TAPE-STATUS	Display information on the specified tapes, the monitoring of those tapes, and the associated devices
SHOW-USER-ATTRIBUTES	Display data in JOIN file entry
STATUS	Display information on the status of current jobs
SYSTATUS	Display information on the allocation of system files and the TASKLIB object module file on the individual procedure levels



## Commands for transaction mode (DCAM)

Command	Brief description of function
APPLICATION	Substitute current values for the application name, the distribution code name, and various passwords in a DCAM program. These values apply as long as the DCAM is running
CONNECTION	Substitute current values for the names of the partner, the processor node and user field, and for a password in a DCAM program. These values apply as long as the connection remains established

## Commands for operation of terminals

### Batch terminals

Command	Brief description of function
RSTART	Activate an inactive batch terminal
RLOGON	Identify the batch terminal user to the system
RJOB	Assign names for batch jobs and control the job output mode
RMSG	Send messages during batch processing
RSTATUS	Interrogate the status of remote batch jobs in batch mode
ROUT	Request deferred job output in batch mode
RLOGOFF	Terminate batch mode
RSTOP	Disconnect a batch terminal from the operating system
PRINT	Output files to a batch terminal (operand DEVICE=REMOTE)



With the exception of PRINT, the commands listed for batch terminals are supported under BS2000 V10.0A for the last time.

## Terminals

Command	Brief description of function
TCHNG	Change terminal's logical properties

## SPOOL jobs

Command	Brief description of function
PRINT	Output files to printer or magnetic tape
PUNCH	Output files to floppy disk or POOLER tape
SHOW-SPOOL-CHARACTER-SETS	List the character sets defined with the system administrator command ADD-SPOOL-CHARACTER-SET
SHOW-SPOOL-DEVICES	List the devices specified with the system administrator command ADD-SPOOL-DEVICE
SHOW-SPOOL-FORMS	List the forms specified with the system administrator command ADD-SPOOL-FORM
SHOW-SPOOL-PARAMETERS	Output the current SPOOL parameter list

## Commands for job variable functions

Command	Brief description of function
CATJV	Create a job variable and modify its attributes
DCLJV	Define a job variable link name or create a job variable
DELON	Cancel the effect of the previously specified ON command
ENDON	Delimit an ON or timeout statement sequence
ERAJV	Erase a job variable
GETJV	Read the value of a job variable
MODIFY-JV- CONDITIONALLY	Read, set, or compare a job variable within a procedure
ON	Cause a job to wait asynchronously, a number of times during a certain period, until a specified condition is satisfied.
SETJV	Set the value of a job variable
SHOW-JV-LINK	Output allocations between job variable names and their file link names to SYSOUT
SKIPJV	Branch within a command sequence, depending on the interpretation of a condition formulated as an operand
STAJV	Interrogate the attributes of a job variable
SHOW-CJC- STATUS	Interrogate conditional job control (CJC)
WAIT	Place a current job in the wait state until a specified condition is fulfilled or until a predefined period of time has expired

These commands are only available to users with the "JV" software product.

## Commands for multiprocessor system users

Command	Brief description of function
MRSSTA	Interrogate the structure of the current multiprocessor system network from the point of view of the local computer
STAM	Output the MRSCAT entries to SYSOUT

## Commands for remote file access (RFA)

Command	Brief description of function
RFAEND	Clear down the RFA connection and terminate the AFR partner job
RFASTART	Set up the RFA connection and generate an AFR partner job

## Commands for file transfer (FT)

Command	Brief description of function
CANCEL-FILE-TRANSFER	Terminate a file transfer (FT) job
SHOW-FILE-TRANSFER	Interrogate FT jobs
TRANSFER-FILE	Initiate a file transfer job

These commands are only available to users with the software "FT" product.

## Commands for messages

Command	Brief description of function
HELP	Request explanatory text on system messages
MODIFY-MSG-ATTRIBUTES	Define language for message output and scope (system-wide, task-specific)
MRSCONTROL	Add task-specific message files to the message system
SHOW-MSG-DEFAULTS	Display information about the number, names, language and scope of message files

# User commands

This chapter contains detailed descriptions of the BS2000 user commands.

## Organization of the command descriptions

The commands are described in alphabetical order by name. They can easily be located using the running headers in the upper corner of the page. As a rule, the command descriptions are organized as follows:

- command name and abbreviated form
- application area, command environment
- command description: command function
- format and operand descriptions
- example: example of function + explanation

## Rules for abbreviating command input

The following can be abbreviated:

- command names
- operand names
- keywords

Each of the commands listed in this chapter can be truncated from right to left as long as it remains unambiguous. This applies to all commands appearing in the ISP command list of the system. Operand names and keywords need only be unambiguous within the command concerned. Ambiguities may cause an error message to be issued.

Default value

Defaults values are provided for many of the frequently used operands in commands. On command execution, a default value is always used if you fail to specify the operand in question when entering the command, or use the null string (null input) as the value for the operand.

Default values are either specified in the command format (frequently encountered among alternative keyword operands) or they must be taken from system tables (e.g. "catid", "userid" for the positional operand "pathname" or default values for several of the PRINT command operands). Specification of the null string as the operand value has a special significance in the case of the FILE command. See page 211.

Example

The CATALOG command provides an alternative default value for each keyword operand when creating a catalog entry. The default value is always used when an operand is omitted or when the null string is specified as the value for the operand.

```
(IN) CATALOG DATA, STATE=, ACCESS=, SHARE=, LARGE=, DESTROY=
```

Check the catalog entry:

```
(IN) FSTATUS DATA, ALL
(OUT) 00000000 :N:$ABC123.DATA _____ (01)
      FCBTYP = NONE VSNTYPE = NONE
      SHARE = NO ACCESS = WRITE
      ACL = NO AUDIT = NONE DESTROY = NO
      CRDATE = NONE EXDATE = NONE LADATE = NONE
      RDPASS = NONE WRPASS = NONE EXPASS = NONE
      ACCESS# = 000 VERSION = 000
      LARGE = NO BACKUP = A MIGRATE = ALLOWED
      BLKTYPE = NONE BLKSIZE = 000000 BLKCTRL = NONE
      RECFORM = NONE RECSIZE = 000000
      VSN/DEV/EXT = NONE
      :N: PUBLIC: 1 FILE RES= 0 FREE= 0 REL= 0 PAGES
```

(01) The file attributes have been determined in accordance with the default values specified in the description of the CATALOG command, both for the operands omitted and the operands specified with null input.



## Job, task, process

BS2000 distinguishes between the concepts "job", "task" and "process". These terms are used to describe certain statuses and activities associated with the functions of the operating system.

**Job:** A sequence of commands, statements and data between LOGON and LOGOFF. A distinction is made between batch jobs and interactive jobs. In a batch job, the string of commands, statements and data is read from a file; in an interactive job, the string is entered from the terminal.

A job is assigned to a job class by the Job Management System and put in the appropriate queue. The job is then given a job number (TSN), which can be used to reference it for the period of time it is resident in the system.

```

/LOGON
:
:
Sequence of commands,
statements and data
:
:
/LOGOFF

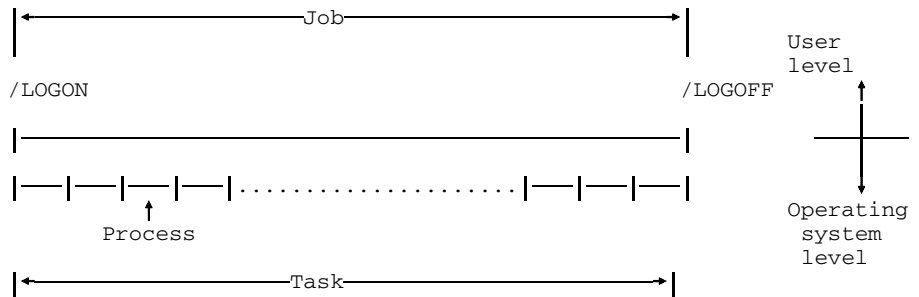
```

} Job

**Task:** From the point of view of the operating system, a job becomes a task when system resources (CPU, memory, devices) are assigned to it. The task is managed by the Task Management System; a task control block (TCB) is created.

**Process:** The activities at program or module level which execute within the task are known as the processes of the task. Each process has its own process control block (PCB), which is used for recording the context of the program in the event of program interrupts. The processes of a task are managed with the aid of the task control block.

The following diagram serves to illustrate the relationship between the terms "job", "task" and "process".



## Job class

A job class is assigned to each job by the Job Management System. Jobs of a particular class share common job attributes, e.g. maximum CPU time, possibly the start time, maximum job and run priority, etc. The Job Scheduler selects the jobs to be started. Each job class is (temporarily) assigned to a Job Scheduler, which in turn is managed (i.e. called) by a Job Stream. See the manual "System Administrator's Guide" [1] for more details.

## Names

Every file cataloged in BS2000 is unequivocally identified by what is known as a path name composed of the file name assigned by you and a "prefix" assigned by the system and corresponding to the access path in the MPVS.

### Path name

The path name is the BS2000 file name entered in the user catalog. It includes the catalog ID and the user ID and uniquely identifies a permanent or temporary file, a file generation or a file generation group.

A path name has the following format:

pathname

:catid:\$userid.filename;  
total length: up to 54 characters

catid

catalog ID; identifier of the pubset in which the file is cataloged;  
Length: 1 to 4 characters;  
Default catid : the catalog ID assigned to the user ID in the JOIN entry; i.e. the identifier of the pubset in which your files are stored by default.  
The catalog ID is always specified between colons (:catid:).

userid

user ID; length: up to 8 characters;  
Default user ID: usually the user ID for the current job, i.e. the user ID entered with the LOGON command; When referring to files, the user ID is always specified with a prefixed "\$" (*\$userid*).

filename

A fully-qualified file name (see below) defined by you. Length:

- for permanent files, up to 38 characters (with 4-character catid) or 41 characters (with 1-character catid)
- for temporary files, incl. prefix, up to 28 characters (with 4-character catid) or 31 characters (with 1-character catid)
- for file generation groups, up to 31 characters (with 4-character catid) or 34 characters (with 1-character catid)

## Fully-qualified file name

A fully-qualified file name is the file name assigned by the user and is linked with the default catalog ID and the default user ID and is then entered in the file catalog. Periods can be used to split a fully-qualified file name into further parts (subnames).

### *Format*

---


$$\text{name}_1[.\text{name}_2[\dots]] \left\{ \begin{array}{l} (*\text{absgen}) \\ (\pm\text{rel}) \\ (\text{version}) \end{array} \right\}$$


---

### Total length:

Permanent files	up to 38 characters (with 4-character catid) or 41 characters (with 1-character catid)
Temporary files	up to 28 characters (with 4-character catid) or 31 characters (with 1-character catid) (including prefix in each case)
File generation groups	up to 31 chars. (with 4-character catid) or 34 characters (with 1-character catid)
File generations	max. length of the name of a file generation group (see above) plus 7 characters for the absolute generation number

$\text{name}_1$   
subname-1 (see below: "File name format conventions")

$\text{name}_2$   
subname-2 (see below: "File name format conventions")

**absgen**  
absolute generation number:  $0 \leq \text{absgen} \leq 9999$  - the generation number identifies file generations within a file generation group.

**rel**  
specifies the relative generation number. Any integer between 1 and 99 is permissible for "rel", which refers to a basic value (see BASE field in FSTAT command output). The following applies:  
rel = abs - basic value. "rel" identifies the relative expression for the file generation as compared with other generations within the group.

**version**  
version identifier for tape files (see below); short name comprising one or more characters (letters, digits or special characters).

*File name format conventions*

You can use the following characters:

- all letters (A ... Z)
- all digits (0 ... 9)
- special characters #, @, \$; hyphen (-) and period (.)

The fully-qualified file name must contain at least one letter, which may be located in some position within the whole character sequence.

A period divides a fully-qualified file name into subnames (see above name<sub>1</sub>.name<sub>2</sub>). Therefore, a period must not be used as the first or last character of a subname.

A hyphen must not be used as the first or last character of a subname.

The character "\$" has a special function as identifier for the user ID. For this reason it must not be used as the first character of a file name.

The characters "@" and "#" have a special function as prefix to identify temporary files. Therefore, they must not be used as the first character of a file name.

## Wildcards in the catalog ID and file name

Wildcard	Meaning
*	Replaces any sequence of characters, including a blank sequence.
/	Replaces any one character exactly.
<wildcard1,...>	Replaces all character sequences to which one of the specified wildcards applies.
<wildcard1,wildcard2>	Replaces a character sequence which: <ul style="list-style-type: none"> <li>- is at least as long as the shortest wildcard character sequence</li> <li>- is no longer than the longest wildcard character sequence</li> <li>- alphabetically, is located between "wildcard1" and "wildcard2"; numbers are sorted after letters.</li> <li>- "wildcard1" may also be a series of blanks. When sorted alphabetically, it is placed in the first position.</li> </ul>
<wildcard1:wildcard2>	Wildcards of the type "wildcard1:wildcard2" may also be specified in list form. The above conventions apply to every such range specification. The system performs a logical OR operation - i.e. the wildcard list replaces all character sequences to which one of the range specifications applies. The length characteristics apply on a pair-by-pair basis, i.e. in each case for a range specification of the type "wildcard1:wildcard2" - not for the entire list.
-wildcard	Replaces all character sequences which do not match the specified wildcards. The minus sign may only occur at the beginning of the wildcard character sequence.

## ABEND Terminate current job

Application group: Job control (page 22 ff.)

### Command description

The ABEND forces the current job to abort. A user memory dump (memory classes 5 and 6) is output if

- OPTION DUMP=YES has been specified previously, or if
- the DUMP prompt was answered with YES (OPTION DUMP=STD).

For the purposes of the dump, a disk file is initialized with the name "\$userid.DUMP.jname.tsn.ser" (where tsn = TSN of the terminated job, ser = serial number).

Otherwise, ABEND functions in the same manner as the LOGOFF command: the memory pages and devices assigned to the task are released and the system asks whether system (output) files are to be output to printer or tape.


The system files SYSLST and SYSOUT are not printed out if they are empty when the ABEND command begins to execute.

If an ABEND command is specified while a program is loaded, any ABEND-STXIT routine that has been defined is activated unless the ABEND command is called via the CMD macro (see the "Executive Macros" manual [5]).

Any monitoring job variable for the job to be terminated is set to "\$A".

**Format and operand description**

Operation	Operands
ABEND	[BUT] [, TAPE] [TAPE] [, BUT] [NOSPOOL] [, BUT] [BUT] [, NOSPOOL]

- BUT** Only works in interactive mode. You enter it to indicate that you wish to start a new job after terminating the current one, and that therefore the terminal should not be disconnected from the host. If BUT is omitted, the terminal will be disconnected from the host.
- TAPE** Causes the system files to be output to tape rather than to printer. The files SYSLST/SYSOUT are written to the same tape in a file with the name "TAPE.TSNnnnn", where "nnnn" is the TSN of the job being terminated by the ABEND command.
- The SYSOPT file is written to a separate tape and is likewise given the file name "TAPE.TSNnnnn", where "nnnn" is a new TSN. You are informed of this via the SYSOUT file.
-  TAPE is only supported for reasons of compatibility. The PRINT or PUNCH command is used to spool out system files to tape.
- NOSPOOL** Suppresses output of system files SYSLST and SYSOUT to printer.



# ABORT      Abort procedure

Application group: Job control (page 22 ff.)

### Command description

The ABORT command enables the user to interrupt a procedure. Procedure mode is terminated at the point where the ABORT command is issued and SYSCMD returns to primary command input (cf. the ENDP command: "Return to the preceding procedure level", see also example 2). All the system files (including TASKLIB) which were opened during procedure execution are closed and their primary allocations are restored.

### Format and operand description

Operation	Operands
ABORT	

The ABORT command is specified without operands.

## Examples

### *Example 1*

Fig. ABORT-1 ABORT and ENDP commands at procedure level 1

The arrows indicate the path of SYSCMD and SYSDTA. At procedure level 1, the ABORT command is identical to the ENDP command.

*Example 2*

Fig. ABORT-1 ABORT and ENDP commands at procedure level 3

The arrows indicate the path of SYSCMD and SYSDTA. The ABORT command results in a branch to the primary command input; an ENDP command at this point would return to procedure level 2.

## ADD-ISAM-POOL-LINK      Define ISAM pool link name

Application group: File processing (page 26 ff.)

### Command description

The ADD-ISAM-POOL-LINK command allows a task-specific pool link name to be assigned to an ISAM pool. The pool link name is entered in the table of pools.

Unequivocal identification of an ISAM pool is possible only by means of its name, catalog ID (host system) and its scope.

When a file is opened, a check is carried out as to whether a pool link name has been specified for it and whether an ISAM pool exists for that link name. The pool link name is entered both in a table and in the TFT. Entry in the TFT is by means of the FILE command.

### Format and operand

Operation	Operands
ADD-ISAM-POOL-LINK	LINK-NAME=name  , POOL-NAME=name ( [CAT-ID= <u>{*DEFAULT-PVS}</u> {catid} ] [ ,SCOPE= <u>{TASK}</u> {HOST-SYSTEM} ] )

- LINK-NAME      Link name by means of which the ISAM pool can be accessed.
- =name      Pool link name. Length: 1-8 alphanumeric characters. The first character must be a letter or one of the characters "\$", "#", "@".
- POOL-NAME     Name of the ISAM pool to which the link name is to be assigned.
- =name(...) Pool name.
- CAT-ID     Catalog ID of the pubset to which the ISAM pool is to be assigned.
- =\*DEFAULT-PVS      The catalog ID assigned to the user ID is used; default value
- =catid      Catalog ID of the pubset.
- SCOPE      Scope of the ISAM pool.
- =TASK      The ISAM-Pool can be used by the user-own task only; default value.
- =HOST-SYSTEM      The ISAM pool is available to all tasks.

## APPLICATION      Enter DCAM application in CLT

Application group:      Transaction mode (DCAM) (page 33)

### Command description

The APPLICATION command is intended for DCAM users; it permits information relating to a DCAM application to be stored in or deleted from a task-specific table known as the Communication Link Table (CLT).

The DCAM application is opened by a DCAM user program; the values in this table replace the corresponding entries in the application control block ACB (Assembler) or in the application structure (COBOL). The link between the CLT entry and this program area is established by means of the link name, which must be specified both in the command and in the program. (See also "DCAM Program Interfaces" [6] and "DCAM Macros" [17].)

The APPLICATION command can only influence the opening of the DCAM application if a link name has been defined in the program.

### Format and operanden description

Operation	Operands
<pre>{APPLICATION} {APPL}</pre>	<pre>[application-name] , LINK=linkname [, DISNAME=distribution-name] [, USEPASS=password1] [, USEPW=password2] [, LOGPASS=password3]</pre>

application-name

Specifies the name of the DCAM application. It must not exceed 8 bytes in length and must consist of EBCDIC characters. The first character must be a letter or the character "@" or "#".

LINK=linkname

Specifies a link name of no more than 8 EBCDIC characters, the first of which must be a letter or the character "@" or "#".

This link name establishes the relation between the entry in a process-specific table (CLT) and an area in the program (ACB or application structure).

Special function: If LINK is the only operand in the APPLICATION command, the CLT entry associated with this link name will be deleted.

DISNAME=distribution-name

Specifies the distribution name under which this process is to receive messages. The name must not exceed 8 bytes in length and must consist of EBCDIC characters. The first character must come from the set (A, B, ..., Z, @, #).

USEPASS=password1

Specifies the password which is defined in the primary process on opening a DCAM application and is required for linking a secondary task to a DCAM application.

The password is 4 bytes in length and is either a character constant (C'xxxx') or a hexadecimal constant (X'xxxxxxxx').

USEPW=password2

Specifies the password which must be defined in the secondary process so that it can be linked to a DCAM application. The password is 4 bytes in length and is either a character constant (C'xxxx') or a hexadecimal constant (X'xxxxxxxx').

LOGPASS=password3

Specifies the password which is defined in the primary process on opening a DCAM application and which must be given by the communication partners when establishing a connection. The password is 4 bytes in length and is either a character constant (C'xxxx') or a hexadecimal constant (X'xxxxxxxx').

**Example**

When a DCAM application is opened, the APPLICATION command shown below causes the application name to be changed and one of the passwords, i.e. XYZ, to be defined.

Fig. APPL-1 Entering the DCAM application in the CLT

## AUDIT Log branch instruction addresses

Application group: Job control (page 22 ff.)

### Command description

The AUDIT command allows you to monitor a program run. This means that you can trace the execution of a program retrospectively by using AUDIT to list the addresses of successful branch instructions. The branch instruction addresses are entered into a special table called the AUDIT table. This table contains 64-word entries and, unless arranged otherwise, is cyclically overwritten.

The AUDIT command can be specified for the entire task run or for a process within that task (e.g. contingency process).

This is also possible after the LOAD command.

The AUDIT table is output by means of the the DISPLAY, DUMP or GET operand or by means of the AID command (see also the "AID" manual [23]).






Application of the AUDIT function to systems with IN-HSI (industry standard normal mode hardware/software interface) significantly increases the program runtime.

### Format and operand description

Operation	Operands
AUDIT	$[\text{ACTION} = \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \\ \text{DISC} \end{array} \right\}]$ $[ , \text{SCOPE} = \left\{ \begin{array}{l} \text{FUNCT} \\ \text{TASK} \end{array} \right\}]$ $[ , \left\{ \begin{array}{l} \text{TID} = \text{tid} \\ \text{TSN} = \text{tsn} \end{array} \right\}]$ $[ , \text{SAVE} = \text{n}]$ $[ , \text{DISPLAY} = \text{YES}]$ $[ , \text{DUMP} = \text{YES}]$ $[ , \text{GET} = \text{YES}, \text{TABLE} = \text{X'addr' }]$



ACTION	Activates or deactivates the logging of branch instruction addresses.
=ON	Activates logging. If no AUDIT table exists, a new one is created and initialized by overwriting all fields with binary zeros. The current pointer is set to the beginning of the table, and logging is activated; this is the default value.  If an AUDIT table already exists, the current pointer is set to the next field to be written, and logging is continued from that point.
=OFF	Deactivates logging and releases the contents and storage space of the AUDIT table as well as that of any existing save table (see the SAVE operand).
=DISC	Deactivates logging but retains the AUDIT table and any existing save table (discontinue). DISC can only be used for SCOPE=TASK, where neither the TID nor the TSN operand may be specified.
DISPLAY=YES	Outputs the 256-byte table to SYSOUT. The entries in it appear in reverse chronological order, i.e. the last entry is displayed first.   DISPLAY must not be specified in conjunction with the TID or TSN operand.
DUMP=YES	Outputs the 256-byte table to SYSLST. The entries appear in reverse chronological order, i.e. the last entry is displayed first.   DUMP must not be specified in conjunction with the TID or TSN operand.
GET=YES	Writes the 256-byte table to the memory area identified by the virtual address specified in the TABLE operand. The entries appear in reverse chronological order, i.e. the last entry is displayed first.   GET must not be specified in conjunction with the TID or TSN operand.
SAVE	Creates a save table in the privileged class 5 memory of the task being logged (in addition to the 256-byte table). This save table can be up to 64 Kb in size and increases the amount of AUDIT information that can be recalled accordingly. Entries in the 256-byte table are saved here before being overwritten.  Save tables are also overwritten cyclically and - like foreign task tables - can only be output with the Advanced Interactive Debugger AID or by means of a dump.

The SAVE operand is only effective in connection with ACTION=ON and only if AUDIT was previously deactivated with ACTION=OFF, i.e. if no AUDIT table exists. The size of the save table can thus be changed only via a task-wide deactivation and reactivation of the AUDIT function.

=n	<p>Number of 4-Kb pages defining the size of the save table.</p> <p>The maximum size of the save table is 64 Kb (<math>0 \leq n \leq 16</math>). Default value: n=0, i.e., no save table is generated.</p>
SCOPE	<p>Indicates the program area to be logged. The area to be be logged (AUDIT command) begins at the position where the program is interrupted by an (interrupt) command and the AUDIT command is issued. Logging begins at the point the program run is continued (RESUME command) and covers the entire (remaining) program run (SCOPE=TASK) or only the process which has just been interrupted (SCOPE=FUNCT). In the latter case, it is the basic process or a contingency process.</p> <p>An AUDIT job for a larger program area or a higher PCB replaces that of a smaller program area or a lower PCB (TASK replaces FUNCT etc.), but not vice versa.</p>
=FUNCT	<p>Specifies that only the process which has just been interrupted is to be logged. FUNCT is only allowed for the user-own task.</p>
= <u>TASK</u>	<p>Indicates that the requesting task or the task specified in the TID or TSN operand is to be logged; default value.</p>
TABLE	<p>Specifies the address of the area to which the contents of the AUDIT table are to be copied. The TABLE operand is ignored if GET=YES is not specified.</p>
=X'addr'	<p>Virtual memory address consisting of 1 to 8 hexadecimal digits. The address must refer to an assigned memory area for which write access is authorized (only for P1). The address "0" is invalid. Only the last six digits are relevant in 24-bit addressing mode; in 31-bit addressing mode, all eight digits are relevant. The addressing mode of the program to be logged applies.</p>
TID	<p>Only valid for SCOPE=TASK. Indicates the task to be logged.</p>
=tid	<p>h[hhhhh] Internal task number (1-8 hexadecimal digits, padded to 8 positions by the system with leading zeros if required).</p>


TSN Only valid for SCOPE=TASK. Indicates the task to be logged.

=tsn Job number (TSN). It may be supplied in any of the following forms:

n[nnn]: 1-4 digits, padded to a 4-digit number by the system with leading zeros if required.

a[aaa]: 1-4 alphanumeric characters, padded by the system to a 4-digit hexadecimal number with leading zeros as required.

C'a[aaa]': 1-4 characters, padded by the system to four positions with leading zeros if required (leading blanks entered by the caller are retained).

 If neither the TID nor TSN operand is specified, the requesting task itself is logged.

## BCNTRL Suppress operator messages

Application group: Job control (page 22 ff.)

### Command description

The BCNTRL command allows a user to suppress (for his job) the display of operator messages which are sent by the operator via MESSAGE or BROADCAST commands. The default value for the first call in a task is the operand value YES. Thereafter, on each subsequent call, the default value used will be that of the last operand specified. At least one operand must be specified, however.

Urgent messages such as system shutdown messages will nevertheless be output to SYSOUT.

### Format and operand description

Operation	Operands
$\left\{ \begin{array}{l} \text{BCNTRL} \\ \text{BC} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{MES} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \quad [ , \text{BCST} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} ] \\ \\ \text{BCST} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \quad [ , \text{MES} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} ] \end{array} \right\}$

MES	All operator texts sent for this job by the MESSAGE
=YES	command will be received on SYSOUT. YES is the default value for the first call in a task.
=NO	No texts sent by the MESSAGE command will be output to SYSOUT.
BCST	All operator texts sent by the BROADCAST command will be
=YES	received on SYSOUT. YES is the default value for the first call in a task.
=NO	No texts sent by the BROADCAST command will be output to SYSOUT.

**Example:**

The command

```
/BCNTRL BCST=NO
```

suppresses only those messages which are sent by the operator via the BROADCAST command. Messages sent by the operator via the MESSAGE command are handled according to the default value for MES=....

## **BREAK Request system mode**

Application group: Program control (page 30)

### **Command description**

The BREAK command can only be issued in a batch job or a procedure. It interrupts a program which is waiting for data, and enables command input. The RESUME command may be used to return to program mode. The change of mode is procedure-internal (the procedure level is retained).

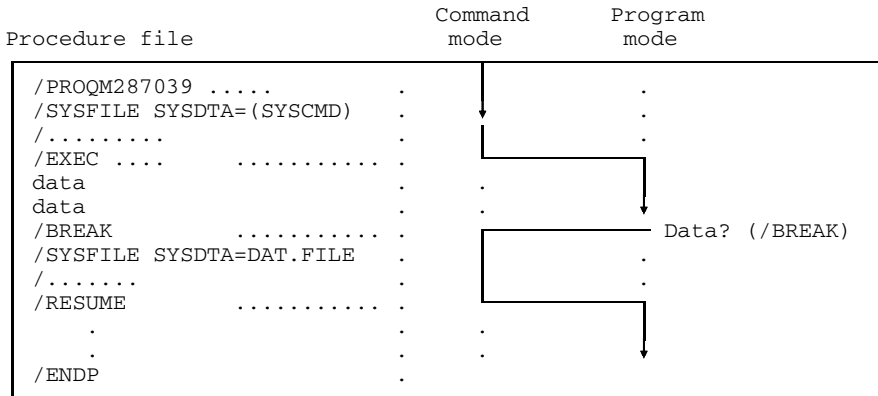


Fig. BREAK-1 Application of the BREAK command

The BREAK command is ignored if entered in system mode; a message is displayed.

### **Format**

Operation	Operands
BREAK	

**Examples:**

In a procedure file, the SYSFILE command is used to assign the (system) files SYSDDTA and SYSCMD to the same file. The U.LOAD program, which reads in data from SYSDDTA, is then called:

Procedure file:

```

/PROC C
/SYSFILE SYSDDTA=(SYSCMD)
/EXEC U.LOAD
123      }
459      } Data for U.LOAD program
/BREAK _____ (01)
/SYSFILE SYSDDTA=U.LOAD.DATA
/RESUME
/ENDP

```

- (01) Data input is interrupted by the BREAK command and control passes to command mode. The SYSFILE command then assigns system file SYSDDTA to the file U.LOAD.DATA. The RESUME command causes the program to be continued. The data for program U.LOAD is then obtained from the file U.LOAD.DATA, as shown in the following runtime log:

Runtime log:

```

(IN)      PROC A
(IN)      SYSFILE SYSDDTA=(SYSCMD)
(IN)      EXEC U.LOAD
(OUT)    % BLS0500 PROGRAM 'RDATA', VERSION ' ' OF '86-05-16' LOADED.
(IN)      123
(OUT)    123      ***** DATA READ IN *****
(IN)      459
(OUT)    459      ***** DATA READ IN *****
(IN)      BREAK
(IN)      SYSFILE SYSDDTA=U.LOAD.DATA
(IN)      RESUME
(OUT)    567      ***** DATA READ IN *****
(OUT)    489      ***** DATA READ IN *****
(IN)      ENDP

```

## CALL Initiate procedure file

Application group: Job control (page 22 ff.)

### Command description

The CALL command is used to initiate a command sequence stored in a file (procedure file). The procedure file is opened, and the command sequence is then processed. During execution, the symbolic operands contained in the file are replaced with the current operand values specified in the command call. This operation is called a CALL procedure.

Procedure files are SAM or ISAM files with variable record length (RECFORM=V). A procedure file is stored

- as a cataloged (also temporary) file,
- as a J-type member in a program library (LMS library routine).

Passwords (CATALOG passwords) can be used to protect procedure files from unauthorized reading (RDPASS), overwriting (WRPASS) and execution (EXPASS). The EXPASS password or a higher-ranking password must be specified by means of the PASSWORD command before the CALL command is issued.

If a library member is used as a procedure file, a temporary SAM file is created. This file contains the following member:

S.IN.library.member.tsn.date.hhmmss

where:

- library = library name (max. 20 characters are printed)
- tsn = job TSN
- date = date
- hhmmss = time-of-day in hours-minutes-seconds

This temporary file is automatically deleted when the following commands are used:

- /LOGOFF in the ENTER file
- /ENDP in the DO or CALL command
- If the CMD macro is used to invoke the CALL command, the calling program is unloaded. Any ABEND-STXIT routine defined in the program is not activated (CMD macro, ABEND-STXIT routine, see "Executive Macros" manual [5]).

The first record in a procedure file is the PROCEDURE command. Subsequent records include commands or data. The last record is the ENDP command (Fig. CALL-1). The PROCEDURE, LOGON or RESTART must not be specified between these records. Otherwise the system issues an error message and processes the next command.



Symbolic operands (e.g. "&X", see example 3) can be used in any position in all the procedure files commands. Current operand values are also substituted for the symbolic operands in the TYPE, PAUSE and REMARK commands. When the processed procedure records are logged to the SYSOUT system file, symbolic operands are replaced by the current operand values.

When the CALL command is issued, the procedure file is assigned to the (system) file SYSCMD. If data is also to be read from the procedure file, you must equate the system files SYSDTA and SYSCMD in the procedure file. [Command: /SYSFILE SYSDTA=(SYSCMD)].

A procedure file with data records must contain the /SYSFILE SYSDTA=(SYSCMD) command (before the data) (see SYSFILE command).

It is not necessary to repeat this assignment within a nesting.

In a batch job, any existing assignment at procedure level 0 is also included in the next procedure level.

## **Nesting of procedures**

Any command procedures can be nested.

Nesting means:

Processing of the commands in a procedure file is interrupted by another procedure file call. Once the ENDP command has been reached, the remaining commands of the last procedure file to be left are processed.

The difference between the CALL procedure and the DO procedure is in the way nested commands are processed (Fig CALL-1): once the ENDP command has been reached, the next command to be read is the one following the last CALL command. This is also applicable when DO and CALL procedures are mixed.

Fig. CALL-1 Nesting of CALL procedures

## Format and operand description

Operation	Operands
CALL	pathname [ , { sympar= [paramvalue] } , ... ] [ sympar= ] paramvalue

pathname	stands for: [ :catid: ] [ \$userid. ] { filename library (member) }
catid	Catalog ID of the pubset on which the file is stored. Default value: the catalog ID assigned to the user ID (JOIN entry).
userid	User ID to which the file is assigned. Default value: the user ID from the LOGON command.
filename	Name of a cataloged file (the procedure file). "filename" can also be the name of a temporary file (see also the manual "DMS Introductory Guide and Command Interface" [8]). The name of a file generation or file generation group may not be specified.
library	Name of an LMS disk library containing the member "member" (see also the "LMS" manual [14]).
(member)	Name of a (J-type) member in the library. "member" consists of up to 38 characters: <ul style="list-style-type: none"> <li>– alphabetical characters           A, ... Z</li> <li>– special characters               \$, #, -, @</li> <li>– numeric characters               0, ... 9</li> </ul> The first character must be a letter. The last character must not be a hyphen. <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-top: 5px;">!</div> The expression "library(member)" can be up to 41 characters long (including parentheses). "library" or "member" can be up to 38 characters long.
sympar	Name of a symbolic keyword operand declared in the PROCEDURE command of the procedure file, but without the initial character "&".

paramvalue

Current value of a keyword operand declared in the PROCEDURE command (up to 254 characters long) or a positional operand.

The current operand values "paramvalue" from the CALL command replace the symbolic operands in the PROCEDURE command as follows:

- **Keyword operands** are given the current value from the operand list of the CALL command. If this value is missing, the operand value specified in the PROCEDURE command is used. If this value is also missing or if the keyword was specified in the CALL command without a value, the value is requested interactively, but not before the procedure is run. This last possibility is known as "prompting".

These three possibilities are shown in the following table:

		Entry in PROCEDURE command	
		&X=ABC	&X=
Entry	X=	Prompting	Prompting
in	X=DEF	DEF	DEF
CALL cmd.	none	ABC	Prompting

- **Positional operands** receive in succession the operand values from the CALL command which are specified without the "sympar=" keyword. If the current value of a positional operand (indicated by commas) is missing in the CALL command, it is requested interactively during the procedure run (prompting), provided the value is required at all.

Character strings enclosed in single quotes are also allowed as current operand values. Any single quotes in these strings must be entered twice, one after the other.

An empty string (null string) is interpreted as a missing operand value; in interactive mode this causes a request for the operand value during procedure execution.

If the current value of a parameter is requested during procedure execution, and the user presses the K2 key, the system will respond with "Terminate procedure ? Yes/No". If "Yes" is specified, the operating system will insert an ABORT command and terminate the procedure; specifying "No" will repeat the request for the current value of a parameter.

If the question "Terminate procedure ? Yes/No" is answered with "Yes" while a program is loaded, the inserted ABORT will be interpreted as a program statement and result in an error message with regard to the program run. However, this will not cause the procedure to be terminated.

The list of keyword operands and/or positional operands may be enclosed in parentheses (compatibility).

If a command to be executed is to be passed, the slash must already be in the procedure file.

## Examples

### Example 1

The following commands are contained in a file named PROC.1:

```

/PROC C
/ERASE *
/STEP
/SYSFILE SYSDTA=PRIM.ASS
/EXEC $ASSEMB
/ENDP
```

If the command sequence in this procedure file is initiated with the command /CALL PROC.1, the commands are processed in the specified order.

The following listing results:

```

(IN)      PROC C
(IN)      ERASE *
(IN)      STEP
(IN)      SYSFILE SYSDTA=PRIM.ASS
(IN)      EXEC $ASSEMB
(OUT)    % BLS0500 PROGRAM ASSEMB, VERSION 291 OF 84-09-10 LOADED.
(OUT)    V29.1A01 OF SIEMENS BS 2000 ASSEMBLER READY
(OUT)    GIVE ASSEMBLER OPTIONS !
(OUT)    PROCEDURE LEVEL NUMBER 1
(OUT)    SYSDTA : PRIM.ASS
(OUT)    FLAGS IN 0000 STATEMENTS, 000 PRIVILEGED FLAGS, 000 MNOTES
(OUT)    HIGHEST ERROR-WEIGHT : -
(OUT)    SYSTEM MACROLIBRARY : MACROLIB
(OUT)    ASSEMBLY TIME :      1.3308 SEC.
(IN)      ENDP
```

### Example 2

```

/PROC C, (&N), SUBDTA=&
/ERASE *
/STEP
/SYSFILE SYSDTA=&N
/EXEC $ASSEMB
/ENDP
```

} A symbolic parameter &N (positional operand) is defined in procedure file PROC.2.

a) The value of the parameter &N is defined when the procedure is called by means of the CALL command:

```
(IN)      CALL PROC.2, (TRE.PRIM.ASS)
```

Runtime log:

```
(IN)   PROC C, (&N)
(IN)   ERASE *
(IN)   STEP
(IN)   SYSFILE SYSDTA=TRE.PRIM.ASS
(IN)   EXEC $ASSEMB
(OUT)  % BLS0500 PROGRAM ASSEMB, VERSION 291 OF 84-09-10 LOADED.
(OUT)  V29.1A01 OF SIEMENS BS 2000 ASSEMBLER READY
(OUT)  GIVE ASSEMBLER OPTIONS !
(OUT)  PROCEDURE LEVEL NUMBER 1
(OUT)  SYSDTA : PRIM.ASS
(OUT)  FLAGS IN 00000 STATEMENTS, 000 PRIVILEGED FLAGS, 000 MNOTES
(OUT)  HIGHEST ERROR-WEIGHT : -
(OUT)  ASSEMBLY TIME :          1.3327 SEC.
(IN)   ENDP
```

A CALL command with the positional operand enclosed in single quotes has the same effect:

```
/CALL PROC.2, ('TRE.PRIM.ASS')
```

- b) The value of the parameter &N is not defined in interactive mode until the procedure is being executed:

```
(IN)   CALL PROC.2
```

Runtime log:

```
(IN)   PROC C, (&N)
(IN)   ERASE *
(IN)   STEP
(IN)   SYSFILE SYSDTA=&N
(OUT)  &N=
(IN)   PRIM.ASS
(IN)   SYSFILE SYSDTA=PRIM.ASS
(IN)   EXEC $ASSEMB
(OUT)  % BLS0500 PROGRAM ASSEMB, VERSION 291 OF 84-09-10 LOADED.
(OUT)  V29.1A01 OF SIEMENS BS 2000 ASSEMBLER READY
(OUT)  GIVE ASSEMBLER OPTIONS !
(OUT)  PROCEDURE LEVEL NUMBER 1
(OUT)  SYSDTA : PRIM.ASS
(OUT)  FLAGS IN 00000 STATEMENTS, 000 PRIVILEGED FLAGS, 000 MNOTES
(OUT)  HIGHEST ERROR-WEIGHT : -
(OUT)  SYSTEM MACROLIBRARY : MACROLIB
(OUT)  ASSEMBLY TIME :          1.3319 SEC.
(IN)   ENDP
```

*Example 3*

```

/PROC C, (&COMP=ASSEMB)
/ERASE *
/STEP
/SYSFILE SYSDTA=U.TEST
/EXEC $&COMP
/ENDP

```

} In procedure file PROC.3  
a symbolic parameter &COMP  
(keyword parameter) is defined  
as having the default value ASSEMB.

- a) The following CALL command renders the default value for the &COMP parameter effective:

```
/CALL PROC.3
```

- b) The following CALL commands substitute the default value ASSEMB for FOR1:

```

/CALL PROC.3, (COMP='FOR1')   or
/CALL PROC.3, (COMP=FOR1)

```

- c) The value of &COMP is requested interactively at procedure execution:

```
/CALL PROC.3, (COMP=)
```

The listings generated correspond to those in example 2.

*Example 4*

In procedure file PROC.4, a parameter list with positional and keyword parameters is defined:

```

/PROC C, (&N, &COMP=ASSEMB, &ERRFIL=NO, &RD=NONE, &SYMDIC=NO, &LIST=NO, &LOAD), -
/SUBDTA=&
/ERASE *
/STEP
/PARAM ERRFIL=&ERRFIL, SYMDIC=&SYMDIC, LIST=&LIST
/SYSFILE SYSDTA=&N
/EXEC $&COMP
/SYSFILE SYSDTA=(SYSCMD)
/EXEC $TSOSLNK
PROG X, FILENAM=&LOAD, IDA=Y
INCLUDE *
END
/CATALOG &LOAD, STATE=UPDATE, RDPASS=&RD
/ERASE *
/ENDP

```



Three of the many ways in which this procedure file can be called are shown below:

- a) The two positional parameters &N and &LOAD are defined interactively during execution of the procedure; the keyword parameters are assigned default values:

```
(IN)          CALL PROC.4
```

#### Runtime log:

```
(IN)          PROC C, (&N, &COMP=ASSEMB, &ERRFIL=NO, &RD=NONE, &SYMDIC=NO, - —— (01)
              &LIST=NO, &LOAD), SUBDTA=&
(IN)          ERASE *
(IN)          STEP
(IN)          PARAM ERRFIL=NO, SYMDIC=NO, LIST=NO
(IN)          SYSFILE SYSDTA=&N
(OUT)         &N=
(IN)          PRIM.ASS
(IN)          SYSFILE SYSDTA=PRIM.ASS
(IN)          EXEC $ASSEMB
(OUT)         % BLS0500 PROGRAM ASSEMB, VERSION 291 OF 84-09-10 LOADED.
(OUT)         V29.1A01 OF SIEMENS BS 2000 ASSEMBLER READY
(OUT)         GIVE ASSEMBLER OPTIONS !
(OUT)         PROCEDURE LEVEL NUMBER 1
(OUT)         SYSDTA : PRIM.ASS
(OUT)         FLAGS IN 00000 STATEMENTS, 000 PRIVILEGED FLAGS, 000 MNOTES
(OUT)         HIGHEST ERROR-WEIGHT : -
(OUT)         SYSTEM MACROLIBRARY : MACROLIB
(OUT)         ASSEMBLY TIME :          1.5334 SEC.
(IN)          SYSFILE SYSDTA=(SYSCMD)
(IN)          EXEC $TSOSLNK
(OUT)         % BLS0500 PROGRAM TSOSLNK, VERSION 19.0C00 OF 85-05-08 LOADED.
(IN)          PROG X, FILENAM=&LOAD, IDA=Y
(OUT)         @LOAD=
(IN)          LOAD.ASS
(OUT)         % LNK0500 PROG BOUND
(IN)          CATALOG LOAD.ASS, STATE=UPDATE, RDPASS=NONE
(IN)          ERASE *
(IN)          ENDP
```

- (01) The continuation character "-" can be located anywhere provided the system parameter SSMCOPT=Y was set at system generation time. If SSMCOPT=N was set, the continuation character must be in column 72.  
The declaration @LOAD=LOAD.ASS is not made for the command mode but for the program TSOSLNK.

- b) The linked program &LOAD is to be processed by the Interactive Debugging Aid and receives the name LOAD.IDA:

```
/CALL PROC.4, (,LOAD.IDA,SYMDIC=YES)
```

The name of the source program "&N" is requested at procedure execution time.

- c) The U.ASS10 source program is to be assembled and the linked PRO10 program protected by the read password C'%%'.

```
/CALL PROC.4, (U.ASS10,COMP=ASSEMB,RD=C'%%',PRO10)
```

The read password can, for example, also be declared in the form RD='C'%%'.

*Example 5*

A procedure file FN1 is to call an additional procedure file FN2 (see also the ENDP command).

File FN1

```
/PROC C, (&P1=A)
.....
/CALL FN2, (P2=&P1)
```

File FN2

```
/PROC C, (&P2=C)
/.....
/SYSFILE SYSDTA=&P2
/.....
/ENDP
```

&P2 assumes the value of &P1.

*Example 6*

- a) A procedure file P starts with the following PROCEDURE command:

```
/PROC C, (&P1,&S2=V2,&P3)
```

In the following calls, some parameters have been omitted. Their current values are requested interactively (prompting) during procedure execution.

Calls for procedure file P	Result
/call p, (a1,s2=a2,a3)	no prompting
/call p, (a1,s2=a2,)	prompting for P3
/call p, (a1,,s2=a2)	prompting for P3
/call p, (, ,s2=a2)	prompting for P1, P3
/call p	prompting for P1, P3

- b) The prompt mode can also be used for keyword operands. If procedure file P starts with the PROCEDURE command:

```
/PROC C, (&P1, &S2=,&P3)
```

then prompting is necessary for keyword operand &S2 in instances such as the calls below.

Call for procedure file P	Result
/call p, (a1,a3)	prompting for S2
/call p	prompting for P1, S2, P3

For further examples, see the PROCEDURE, ENDP, ESCAPE and RTI commands.

## CANCEL Cancel job

Application group: Job control (page 22 ff.)

### Command description

The CANCEL command cancels jobs running under the user's own user ID. However, the calling job is not allowed to cancel itself (this is only possible with the LOGOFF or ABEND command). The job to be cancelled is identified by its task sequence number (TSN) or a monitoring job variable.

CANCEL, like LOGOFF and ABEND, releases the resources assigned to the task and outputs SYSLST/SYSOPT to printer.

If the job to be cancelled is terminating or has already terminated, the CANCEL command will be rejected and a corresponding message displayed at the terminal.

The CANCEL command will only cancel a job when it is in the TU state (user state) or is waiting for input of a new command.

With spoolout tasks, it is possible for two or more jobs to have the same task sequence number (TSN), e.g. as the result of a PRINT command for a number of files. In this case, the CANCEL command will affect all jobs with the TSN specified in the command.


A job cannot be cancelled if:

- it is abnormally terminated due to a system error;
- it is in the "permanently pended" state;
- a NCHOLD command was issued for the task (by the operator or system administrator);
- it has already reached the termination stage; e.g. actions following LOGOFF or ABEND command/macro, actions in the user program after the ABEND event has been reported (ABEND-STXIT routine, see the "Executive Macros" manual [5]).

### Format and operand description

Operation	Operands
{ CANCEL } { CAN }	{ tsn } [ , DUMP ] [ , NOSPOOL ] { MONJV=jvname }

---

tsn	TSN of the job to be cancelled. Leading zeros need not be specified.
DUMP	Causes a dump to be output to disk for the job to be cancelled. However, this will only take place if previously specified in the OPTION command by selecting one of the following:  OPTION DUMP=YES    Dump is output. OPTION DUMP= <u>STD</u> Dump prompt is displayed; if the response is "Y", a dump is output. OPTION DUMP=NO    Dump is suppressed.
MONJV=jvname	Supplies the name of a job variable defined in the LOGON or ENTER command for the job to be cancelled.  The status indicator of this job variable is set to "\$A". If a program running within this job is being monitored by a job variable, the status indicator of the job variable performing this function is also set to "\$A".  If "jvname" is not being used for monitoring a job or is not accessible, the command is rejected.   This operand is only available with the JV software product (see the "Job Variables" manual [11])
NOSPOOL	The system files SYSLST, SYSOUT (if MSG=H in LOGON or OPTION command) and SYSOPT are not output to printer/punch.

## Examples

Extract from an interactive job with the job number (TSN) 1MEV:

### Example 1: Dialog from terminal A

```
(IN) STATUS LIST _____
(OUT) NAME TSN TYPE PRI CPU-USED CPU-MAX ACCOUNT#
      ABCDEFG 1MHT 3 DIALOG 0 210 0.2038 9999 ACC12345
      ABCDEFG 1MEV 3 DIALOG 0 210 2.9835 9999 ACC12345
(OUT) % SPS0171 NO LOCAL SPOOLOUT JOB PRESENT
(OUT) % SRO0376 NO RSO JOB OF TYPE=7 PRESENT

(IN) CANCEL 1MHT _____
(OUT) % EXC0119 CANCEL PROCESSING INITIATED FOR TSN '1MHT', USERID 'ID9101'

(IN) STATUS LIST
(OUT) NAME TSN TYPE PRI CPU-USED CPU-MAX ACCOUNT#
      ABCDEFG 1MEV 3 DIALOG 0 210 3.0581 9999 ACC12345
(OUT) % SPS0171 NO LOCAL SPOOLOUT JOB PRESENT
(OUT) % SRO0376 NO RSO JOB OF TYPE=7 PRESENT

***** Display at terminal B ***** _____ (02)

(IN) LOGOFF "JOB CANCELLED"
(OUT) % NRTT201 TASK TERMINATION DUE TO /CANCEL(-JOB) COMMAND
(OUT) % EXC0419 /LOGOFF AT 0901 ON 91-01-11 FOR TSN '1MHT'
(OUT) % EXC0421 CPU TIME USED: 0.2241 , SERVICE UNITS: 000000000002050
```

### Example 2: Cancel own dialog

```
(IN) CANCEL 1MEV
(OUT) % EXC0124 USER CANNOT ABORT HIS OWN TASK WITH /CANCEL-JOB
      OR /CANCEL. CMD REJECTED _____ (03)
```

*Example 3: ENTER with MONJV*

```
(IN) ENTER ENTER.1,MONJV=MONJV _____ (04)
(OUT) % JMS0066 JOB 'ENTERDIA' ACCEPTED ON 90-12-18 AT 09:50, TSN = 0X6B
```

```
(IN) STATUS MONJV=MONJV _____ (05)
(OUT) TSN: 0X6B TYPE: 2 BATCH NOW: 901218.095048
JOBNAME: ENTERDIA PRI: 9 230 SPOOLIN: 901218.0950
USERID: USR123 JCLASS: JCB05000 LOGON: 901218.0950
ACCNB: ACCXYZ12 CPU-MAX: 5000 CPU-USED:000000.2356
REPEAT: NO RERUN: NO FLUSH: NO
MRSCAT: HOLD: NO START: SOON
TID: 0002012B UNP/Q#: 17/012 CMD: WAIT
```

```
(IN) CANCEL MONJV=MONJV _____ (06)
(OUT) % EXC0119 CANCEL PROCESSING INITIATED FOR TSN '0X6B', USERID 'USR123'
```

- (01) Use the STATUS command to obtain information on the jobs currently running under your user ID.
- (02) The following messages are displayed on the terminal associated with interactive job 1MHT ("forced LOGOFF"):

```
(IN) LOGOFF "JOB CANCELLED"
(OUT) % NRTT201 TASK TERMINATION DUE TO /CANCEL(-JOB) COMMAND
(OUT) % EXC0419 /LOGOFF AT 0901 ON 91-01-11 FOR TSN '1MHT'
(OUT) % EXC0421 CPU TIME USED: 0.2241 , SERVICE UNITS: 000000000002050
```

- (03) You cannot use CANCEL to cancel your own job.
- (04) The ENTER initiates batch job 0X6B. This job is to be monitored by job variable MONJV.
- (05) The STATUS command requests information about the batch job monitored by MONJV.
- (06) The CANCEL command terminates the batch job 0X6B monitored by MONJV.

## CANCEL-FILE-TRANSFER Cancel file transfer


Application group: File transfer (FT) (see page 37)

This command is only available with the software product FT.

### Command description

The CANCEL-FILE-TRANSFER command allows the processing of one or more FT jobs to be aborted.

Subsets of existing FT jobs can be selected by specifying selection criteria.

 The following command description covers only the abbreviated command format. (The description is valid for FT-BS2000 V4.0B.) Refer to the "File Transfer" manual [7] for a detailed description with application examples.

### Format (abbreviated format) and operand description

Operation	Operands
$\left. \begin{array}{l} \text{CANCEL-FILE-} \\ \text{TRANSFER} \\ \text{NCANCEL} \end{array} \right\}$	$[\text{TRANS}=\left\{ \begin{array}{l} \text{*ALL} \\ \text{no} \end{array} \right\}]$ $[\text{, SELECT}=\left\{ \begin{array}{l} \text{*OWN} \\ \text{(INIT}=\left\{ \begin{array}{l} \text{*LOCAL, *REMOTE} \\ \text{*LOCAL} \\ \text{*REMOTE} \end{array} \right\} } [\text{, PARTNER}=\left\{ \begin{array}{l} \text{*ALL} \\ \text{name} \end{array} \right\} ] \end{array} \right\}]$ $\rightarrow [\text{, FILE-NAME}=\left\{ \begin{array}{l} \text{*ALL} \\ \text{*LIB (LIB}=\left\{ \begin{array}{l} \text{*ALL} \\ \text{lib} \end{array} \right\} \text{), ELEM}=\left\{ \begin{array}{l} \text{*ALL} \\ \text{member} \end{array} \right\} } \right\} (\text{VERS}=\left\{ \begin{array}{l} \text{*ALL} \\ \text{version} \end{array} \right\} \text{), TYPE}=\left\{ \begin{array}{l} \text{*A} \\ \text{ty} \end{array} \right\} \right\}]$ $\rightarrow [\text{, MONJV}=\left\{ \begin{array}{l} \text{*NONE} \\ \text{jvname} \end{array} \right\}] \text{, } [\text{JV-PASS}=\left\{ \begin{array}{l} \text{*NONE} \\ \text{password} \\ \text{*SECRET} \end{array} \right\}]]$



TRANS	Identifies the FT job to be aborted.
= <u>*ALL</u>	All your FT jobs will be aborted; default value.
=no	FT job number (transfer ID).
SELECT={...}	Permits selection of a subset of FT jobs. All selection criteria specified must be valid for the job to which they apply. Processing of the selected FT jobs is aborted.
<u>*OWN</u>	All your FT jobs are aborted; default value.
INIT	Permits selection of FT jobs started in the local or a remote system. The local system is the system where the CANCEL-FILE-TRANSFER command is issued.
= <u>(*LOCAL,*REMOTE)</u>	Selects all FT jobs started in the local or a remote system; default value.
=*LOCAL	Selects all jobs started in the local system.
=*REMOTE	Selects all jobs started in the remote system.
PARTNER	Permits selection of FT jobs processed in cooperation with the specified partner system.
= <u>*ALL</u>	Jobs are selected without regard to the partner system, if any; default value.
=name	Name of the partner system.
FILE-NAME	Permits selection of FT jobs accessing the specified file/ library. The file/library must be cataloged in the local system.
= <u>*ALL</u>	Jobs are selected without regard to the name of the file/ library; default value.
=*LIB(...)	Refers to FT jobs accessing a particular library. The selection can be further restricted by specifying a library name/member name/member version/member type.
<u>*ALL</u>	Jobs are selected without regard to the library name/member name/member version/member type.
lib	library name
member	member name
version	version name

type	member type
=filename	File name
MONJV	Permits selection of FT jobs monitored by a job variable.
=* <u>NONE</u>	Jobs are selected without regard to the monitoring job variable, if any.
=jvname	Name of the job variable.
JV-PASS	Specifies the password, if any, that protects the job variable.
=* <u>NONE</u>	No password has been defined.
=password	Password protecting the job variable.
=* <u>SECRET</u>	The system prompts the user to enter a password. Display of the password on the screen is suppressed when you enter it.

## CATALOG      **Process catalog entry**

Application groups:

File processing (page 26 ff.)

Device and volume reservation (page 29)

FGG = file generation group; FG = file generation

### **Command description**

The CATALOG command is used to catalog a file or file generation group or to modify the file names and protection attributes for a cataloged file/FGG.

The following can be defined:

- file name
- read/write access allowed
- shareability
- read, write and execute passwords
- write protection
- save frequency and type of save (full etc.) with the ARCHIVE file saving system
- data protection (data destruction)
- access monitoring by means of DMS routines (AUDIT function)
- migration by HSMS permitted or not permitted

The CATALOG command can also be used to create the catalog entry for a temporary (user) file or to convert a temporary file into a permanent file, and vice versa. The following restrictions apply to temporary files:

- The RETPD operand has no effect.
- The following system default values apply: RDPASS=NONE, WRPASS=NONE, EXPASS=NONE, SHARE=NO, ACCESS=WRITE and BACKUP=E; (SHARE=YES for tape files). These values cannot be changed.
- The following applies when recataloging files from temporary to permanent: the file attributes of the temporary files are retained (exception: the default value is assumed for BACKUP).
- When files are recataloged from permanent to temporary, the default attributes for temporary files are used.

Default values are available for all operands when the catalog entry for a file is created (see the description of the operands).

### **File generation groups (FGG)**

If you wish to work with a file generation group (FGG), the group entry must be created before the first generation is cataloged. Unlike files and file generations, which can also be cataloged using the FILE command, the group entry can be only created with the CATALOG command.

Files can be recataloged as file generations providing the file generations are recreated when this is done. However, file generations cannot be recataloged as files.

File generation groups which are stored on private disks and for which no catalog entries exist are called "foreign" file generation groups. If such FGGs are to be cataloged again, the group entry must be created first. For file generation groups on private disks, the operand STATE=FOREIGN can be used for this purpose if the F1 label on the disk contains the group entry. The system then creates the group entry from the information in the F1 label of the private disk specified via the DEVICE and VOLUME operands.

For the importation of a file generation group whose generations are stored on tape or on a private disk whose F1 label does not contain the group entry, the operands FIRST and BASE must be specified in the CATALOG command for the group entry.

## Files on magnetic tapes and tape cartridges

When creating or updating the catalog entries for tape files, you must observe some special features which result from the use of tapes.

When files with standard labels are created, details of the shareability (SHARE), access type (ACCESS) and passwords are transferred from the catalog entry to the file labels. If SECLEV=HIGH (see the FILE command) is specified for a foreign file, details of the access rights are transferred from the file labels into the catalog entry when the file is opened.

Since file labels on a tape cannot be changed without destroying the file (this is a hardware restriction), and the contents of the catalog entry for a file must match the contents of the file labels, the access rights and the expiration date of a tape file cannot be changed by means of the CATALOG command once the file has been opened and closed correctly.

If the tape file was cataloged using the FILE command, the file protection attributes can be modified using the CATALOG command before the file is opened for the first time. These attributes are then transferred without further checking to the file labels when the file is created. In this way, it is possible to define write protection (ACCESS=READ) for a file which is still to be created. The file can then be opened as an output file and created, after which the write protection comes into effective.

If a tape file has been cataloged with FILE, it will be shareable, unless the CATALOG command is specified with SHARE=NO before the file is opened for the first time.

If password protection is specified for a tape file, the label processing routines transfer the passwords to the HDR3 label, without checking them, when the file is created. Similarly, when a file is imported, passwords are transferred from the HDR3 label into the catalog entry.

The passwords are not transferred for a file for which SECLEV=LOW is specified.

If the system administrator selected password encryption when the system was generated, the encryption indicator in the HDR3 label is set to '1' when the file is opened.

The owner of a tape file can bypass checking of the access authorization by specifying SECLEV=LOW.

Format and operand descriptions

Operation	Operands
<p>{CATALOG} {CAT}</p>	<p>pathname1[, filename2]</p> <p>[, STATE={ NEW UPDATE FOREIGN }] [, ACCESS={ WRITE READ }] [, SHARE={ NO YES SPECIAL }]</p> <p>[, WRPASS={ NONE password }] [, RDPASS={ NONE password }]</p> <p>[, EXPASS={ NONE password }] [, RETPD=days] [, MIGRATE={ ALLOWED INHIBIT }]</p> <p>[, BASE=number] [, DISP={ CYCLE REUSE DELETE KEEP }] [, GEN=number] [, FIRST=number]</p> <p>[, AUDIT={ NONE ALL SUCC FAIL }] [, VOLUME=vsfn] [, DEVICE=device]</p> <p>[, BACKUP={ A B C D E }] [, LARGE={ NO YES }] [, DESTROY={ NO YES }]</p> <p>[, BASIC-ACL={ NONE STD OWNER=access-rights (GROUP=access-rights [, { ... }]) OTHERS=access-rights }]</p>

Unless specified otherwise, the descriptions below apply to files as well as the file generations of a file generation group.

**Positional operands**

pathname1	stands for: [:catid:][userid.]filename1
catid	Catalog ID of the pubset on which the file is stored. Default value: the catalog ID assigned to the user ID (JOIN entry).
userid	User ID. Default value: user ID from the LOGON command. Only the user's own user ID is allowed.
filename1	Fully-qualified name of the file for which a catalog entry is to be created or whose file attributes are to be modified. The name of a file generation (see page 44) may be specified only for the purpose of creating its catalog entry (operand STATE=NEW); the file attributes are determined by the FGG.
filename2	New file name for the file "filename1". This operand must be entered without the catalog ID and user ID. The following applies to tape files: The only permissible "filename2" entry is "filename1" provided with a version identifier in parentheses which must be different from the original version identifier, if any. This restriction is a result of tape label processing: individual blocks of the tape file cannot be overwritten for hardware reasons and the file name in the label is compared with the file name in the catalog entry when the file is opened.

*Example:*

```
/CAT TAPE,TAPE(ONE),STATE=UPDATE
/CAT TAPE(120),TAPE(40),STATE=UPDATE
```

**Keyword operands (in alphabetical order)**

ACCESS	Specifies whether only read access or both read and write access is allowed for the file. Tape files: the ACCESS indicator is copied to the HDR3 label by DMS when the file is opened for the first time. The file owner can specify SECLEV=LOW in order to bypass access mode checking for subsequent file accesses (see the FILE command).
=WRITE	Both read and write accesses are allowed (write access implies read access). WRITE is the default value for a file to be newly cataloged. Tape files: HDR3 label → access type = 0
=READ	Specifies that only read access is permitted. Tape files: HDR3 label → access type = 1
AUDIT	Permits the logging of file accesses by system exit routines. CATALOG, ERASE, FILE and OPEN operations can be audited. This file access logging facility can be restricted to particular user IDs by the AUDIT operand in the JOIN command (see the "System Administrator's Guide" [1]).
=SUCC	All successful DMS actions for the file are to be audited.
=FAIL	All unsuccessful DMS actions for the file are to be audited.
=ALL	All DMS actions for the file are to be audited.
=NONE	No auditing. NONE is the default value when creating the catalog entry.
BACKUP	Specifies the frequency with which the file is to be saved with the regular save runs performed by the ARCHIVE file saving system. The function is performed for disk files only. The default value for BACKUP is specified by the system administrator at system generation and is only effective in conjunction with STATE=NEW.
=A	The file is saved in every save run.
=B	The file is saved whenever a save run for files with BACKUP=B/C/D is performed.
=C	The file is saved whenever a save run for files with BACKUP=C/D is performed.
=D	The file is only saved when a save run for files with BACKUP=D is performed.
=E	No automatic saving by ARCHIVE.



---

BASE	<p>For file generation groups only: defines a reference point or base generation for relative generation numbers.</p> <p>Default value: value of the FIRST operand when the FGG is created, or BASE=0 if no FIRST value is specified.</p> <p>If generations of an FGG for which no group entry exists in the catalog or in the F1 label of the disk are imported from a private disk, the BASE operand must be specified in the CATALOG command for the group entry.</p>
=number	<p><math>0 \leq \text{number} \leq +9999</math>; "number" is the absolute generation number of the base generation. When creating a group entry (STATE=NEW), "number" is entered in the catalog as the base value and this base generation becomes the oldest generation (catalog field FIRSTGN). If specified together with STATE=UPDATE, "number" must refer to a generation which is already cataloged; the BASE value in the catalog entry is then updated accordingly.</p> <p><math>-99 \leq \text{number} \leq -1</math>; "number" is the relative generation number of the new base generation with respect to the most recently cataloged generation (base = LASTGN + number). This relative specification of the base value is permitted only together with STATE=UPDATE (cf. examples).</p>
BASIC-ACL	<p>Only for permanent files; defines how specific user classes can access the file (see the manual "DMS Introductory Guide and Command Interface" [8]).</p>
=NONE	<p>The BASIC-ACL indicator in the catalog entry is reset.</p>

=STD A BASIC-ACL entry is created for the file.

The following default values apply for STATE=NEW (default)  
 OWNER=(READ=YES, WRITE=YES, EXEC=YES), GROUP=NO  
 ACCESS, OTHERS=NO ACCESS.

If STATE=UPDATE is specified at the same time, the existing  
 settings for SHARE and ACCESS are transferred as follows:

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

=(...) Defines individual access rights for the specified user classes.

OWNER= Defines how the owner may access the file.

GROUP= Defines how members of the owner's group may access the file.

OTHERS= Defines how members of other groups may access the file.

access rights

mean:

$$\left\{ \begin{array}{l} \text{NO-ACCESS} \\ \left( \left[ \text{READ} = \left\{ \begin{array}{l} [\text{NO}] \\ \text{YES} \end{array} \right\} \right], \left[ \text{WRITE} = \left\{ \begin{array}{l} [\text{NO}] \\ \text{YES} \end{array} \right\} \right], \left[ \text{EXEC} = \left\{ \begin{array}{l} [\text{NO}] \\ \text{YES} \end{array} \right\} \right] \right) \end{array} \right\}$$

NO-ACCESS No access authorization

READ Read access

=NO Not allowed;

=YES Allowed;

WRITE Write access

=NO Not allowed;


=YES Allowed;

EXEC Execute access

=NO Not allowed;

=YES Allowed;

DESTROY	Specifies whether or not the file is to be destroyed on deletion.
=YES	Whenever a file is released or opened for writing, the storage space occupied by the file is overwritten with X'00' (data destruction). Tape files: any data remaining on tape is destroyed when the file is closed.
=NO	When the file is deleted, its storage space is returned unchanged to the system unless data destruction was specifically requested in the ERASE command (DESTROY operand in the ERASE command). NO is the default value when the catalog entry is created. Tape files: any data remaining on tape is not destroyed unless DESTOC=YES was specified in the FILE command.
DEVICE=device	Only for file generation groups on private disks; designates the disk device on which the FGG is to be stored. The permissible entries for "device" are listed in the device table in the appendix.  The VOLUME and DEVICE operands must be specified when – an FGG is created on private disk(s) (STATE=NEW) – an existing FGG on private disk(s) is to be reconstructed (STATE-FOREIGN).  Otherwise, specification of VOLUME and DEVICE results in an error message.
DISP	Only for file generation groups: specifies whether the oldest generations are to be erased and, possibly, their storage space reused when the maximum number of simultaneously existing generations specified with GEN= is exceeded. In the case of generations on tape, only the catalog entry is deleted.
= <u>CYCLE</u>	The oldest existing generation is erased and its storage space on disk or the tapes it occupies is/are released. The LASTGN and FIRSTGN fields in the catalog entry (youngest and oldest existing generations) are updated accordingly; DISP=CYCLE is the default value.

- =REUSE**      The effects of DISP=REUSE vary depending on the type of volume:
- For FGGs on public disks: the oldest generation is erased, its storage space is returned to the system, and the group entry is updated (see DISP=CYCLE).
- For FGGs on private disks: the new generation is created, the oldest generation is erased, and the volume is used for storing the new generation. If the generation which was deleted extended over several volumes, the new generation is cataloged only on the first of these volumes. The group entry is updated accordingly.
-  Since the old generation is not erased until the new generation has been created, insufficient space on the volume can prevent creation of the new generation even though DISP=REUSE is specified.
- For FGGs on tape: the oldest generation is deleted from the catalog and the new generation is created on the tapes thereby released. The group entry is updated accordingly. DISP=REUSE must not be specified for FGGs in file sets.
- =DELETE**      All generations of the FGG are erased and the new generation becomes the oldest generation of a new series. The group entry is updated accordingly.
- =KEEP**      The "superfluous" oldest generations are not erased automatically, but only when a CATALOG command with the operands FIRST and BASE is used to define a new "oldest" and a new base generation, or when a new value is specified for DISP=. As each new generation is created, only the LASTGN field in the group entry is updated.
- EXPASS**      Defines an execute password to safeguard against unauthorized execution of a load module or of a command sequence (procedure or ENTER task) stored in the file. The password must be entered in the task password table by means of the PASSWORD command before the program or command sequence is executed for the first time. (An execute password includes protection against unauthorized read and write access.) EXPASS must not be specified for an FGG.
- =NONE**      No password is declared. NONE is the default value for creating the catalog entry.

---

=password	Execute password; specified as a C-string, X-string or decimal number (see the WRPASS operand). If X'00000000' is specified for "password", it is ignored.
FIRST=number	<p>Only for file generation groups: specifies the absolute generation number (<math>1 \leq \text{number} \leq 9999</math>) of the first file generation to be cataloged. FIRST may thus be specified only when a file generation group is being created (STATE=NEW).</p> <p>The FIRST operand is needed for reconstruction of the catalog entry of a foreign file generation group which is stored on a private volume, and should be used for this purpose only. File generations stored on tape must be cataloged individually using the FILE command (operand STATE=FOREIGN), while generations stored on private disk can be cataloged by means of the IMPORT command or individually by means of the FILE command (operand STATE=FOREIGN).</p>
GEN=number	<p>Only for file generation groups; <math>0 \leq \text{number} \leq 255</math>; specifies the number of FGG generations that may be cataloged at the same time (cf. DISP operand). GEN may be specified for a new (STATE=NEW) or an existing file generation group (STATE=UPDATE). Default value: GEN=0</p> <p>GEN=0: if it is specified in conjunction with STATE=NEW, a "normal" file rather than a file generation group is created; if is specified in conjunction with STATE=UPDATE, GEN=0 is ignored.</p>
LARGE	Determines the type of backup (for disk files only).
=YES	In save runs using ARCHIVE, only those PAM blocks which have been modified since the last backup operation are saved. (This is of particular relevance for large files.)
=NO	The file is saved in its entirety in every save run using ARCHIVE. NO is the default value when the catalog entry is created.

MIGRATE	Only for files and file generations (FGs) on public disks and only relevant in conjunction with HSMS (Hierarchical Storage Management System). The MIGRATE operand can be used to specify whether files or FGs which have not been used for some time may be migrated from the online storage level to a background storage level with a considerably longer access time. (For more information see the "HSMS" manual.) Default value: MIGRATE = ALLOWED for permanent files; MIGRATE = INHIBIT for temporary files.
=ALLOWED	The file or file generation (FG) may be migrated from storage level S0 to level S1 or S2.
=INHIBIT	The file or file generation (FG) must not be migrated.
RDPASS	Defines a read password to safeguard against unauthorized read access to the file. This safeguarding includes protection against unauthorized write access, but not against execution of an object/load module stored in the file or of a command sequence. In the case of an object/load module, both the file itself and the section currently in main memory are protected against unauthorized read access (the LOAD command and the IDA commands DISPLAY and AT are rejected when access is not authorized). The password must be entered in the task password table prior to the first read or write access by means of the PASSWORD command.
=NONE	No password is defined. NONE is the default value when creating the catalog entry.
=password	Defines a read password, specified as a C-string, X-string or decimal number (see the WRPASS operand).
RETPD	Specifies a file retention period in days, during which time the file may only be read, i.e. it cannot be modified or deleted. The retention period can only be defined for files which already exist, i.e. the operand will be ignored if STATE=NEW is specified. The RETPD operand has no effect on tape files.
=days	Indicates a retention period in days (0..32767). The date on which the retention period ends is computed by the system and noted in the EXDATE field of the catalog entry (see the FSTATUS command). Default value: days=0, i.e. the file may be updated immediately.

SHARE	Specifies whether or not the file is shareable. Shareability means that users with a different user ID can also read the file catalog entry and access the file.
=NO	The file is not shareable. NO is the default value when the catalog entry is created. Tape files: HDR1 label → access indicator = 1.
=YES	The file is shareable. Tape files: HDR1 label → access indicator = 2.
=SPECIAL	The file can also be accessed under the \$SERVICE ID. At the same time, SHARE=YES applies.
STATE	Specifies whether a new catalog entry is to be created or an existing catalog entry is to be changed.
= <u>NEW</u>	A new catalog entry is to be created; default value.
=UPDATE	An existing catalog entry is to be updated. A catalog field is not updated unless the corresponding operand is specified explicitly; default values apply to a new catalog entry only.  If the file is password-protected, the catalog entry cannot be updated unless the write password is entered. For details about password protection of files please refer to the manual "DMS Introductory Guide and Command Interface" [8].
=FOREIGN	Only for non-cataloged FGGs on private volumes; a catalog entry is created for the file generation group stored on private disk. If STATE=FOREIGN is specified, the operands DEVICE and VOLUME must also be specified; any further operands are ignored. The values for GEN, BASE and FIRST are taken from the F1 label of the private disk and placed in the catalog entry. All generations of the foreign FGG to be imported must either be indicated to the system individually in a separate FILE command/macro (operands STATE=FOREIGN, DEVICE and VOLUME, one FILE command/macro for each generation!), or they may be added collectively by means of the IMPORT command.

**VOLUME=vsn** Only for FGGs on private volumes; specifies the private disk on which the group entry is kept via its volume serial number ("vsn"). All related generations must also be on private disks and must be created by means of FILE commands.

The VOLUME and DEVICE operands must be specified when

- an FGG is created on private disks (STATE=NEW)
- an existing FGG on private disks is to be reconstructed (STATE-FOREIGN).

Otherwise, specification of VOLUME and DEVICE results in an error message.

**WRPASS** Defines a write password to protect the file against unauthorized write accesses. Read access is allowed.  
If a password has been defined for a file, it must be entered in the task password table prior to the first write access to the file using the PASSWORD command.

**=NONE** No password is defined. NONE is the default value when the catalog entry is created.

**=password** Write password; specified as a C-string, X-string or decimal number.  
C-string: C 'character-constant'; up to 4 characters.  
X-string: X 'hexadecimal-constant'; up to 8 characters.  
Decimal number: decimal-constant; permitted range:  
-2147483648 through 2147483647.

When a CATALOG command is logged to the system file SYSLST or SYSOUT, all passwords are overwritten with the letter "P".



## Examples

### Example 1: Creating the catalog entry for the file FILE.NEW

```
(IN) CATALOG FILE.NEW _____ (01)
(IN) FSTATUS FILE.NEW,ALL
(OUT) 00000000 :N:$USR123.FILE.NEU
      FCBTYPE = NONE          VSNTYPE = NONE
      SHARE   = NO           ACCESS  = WRITE
      ACL     = NO           AUDIT   = NONE          DESTROY = NO
      CRDATE  = NONE        EXDATE  = NONE          LADATE  = NONE
      RDPASS  = NONE        WRPASS  = NONE          EXPASS  = NONE
      ACCESS# = 000         VERSION = 000
      LARGE  = NO          BACKUP   = A           MIGRATE = ALLOWED
      BLKTYE = NONE        BLKSIZE  = 000000      BLKCTRL = NONE
      RECFORM = NONE       RECSIZE  = 000000
      VSN/DEV/EXT = NONE
      :N: PUBLIC: 1 FILE RES= 0 FREE= 0 REL= 0 PAGES
```

### Example 2: Defining a read password for the file FILE.NEW

```
(IN) CATALOG FILE.NEW, RDPASS=P, STATE=UPDATE _____ (02)
(IN) FSTATUS FILE.NEW,ALL
(OUT) 00000000 :N:$USR123.FILE.NEU
      FCBTYPE = NONE          VSNTYPE = NONE
      SHARE   = NO           ACCESS  = WRITE
      ACL     = NO           AUDIT   = NONE          DESTROY = NO
      CRDATE  = NONE        EXDATE  = NONE          LADATE  = NONE
      RDPASS  = YES        WRPASS  = NONE          EXPASS  = NONE
      ACCESS# = 000         VERSION = 000
      LARGE  = NO          BACKUP   = A           MIGRATE = ALLOWED
      BLKTYE = NONE        BLKSIZE  = 000000      BLKCTRL = NONE
      RECFORM = NONE       RECSIZE  = 000000
      VSN/DEV/EXT = NONE
      :N: PUBLIC: 1 FILE RES= 0 FREE= 0 REL= 0 PAGES
```

Example 3: Creating the catalog entry for a file generation group and its file generations

```

(IN) CATALOG DAT.GROUP,GEN=5,RDPASS=P,BACKUP=C,DESTROY=YES _____ (03)
(IN) CATALOG DAT.GROUP(*1) _____ (04)
(IN) CATALOG DAT.GROUP(+1) _____ (05)
(OUT) % DMS05CC FILE NAME ALREADY CATALOGED OR INVALID OPERAND.
      COMMAND NOT PROCESSED
(IN) CATALOG DAT.GROUP,BASE=1,STATE=UPDATE _____ (06)
(OUT) % DMS05CF FILE PASSWORD-PROTECTED. ENTER REQUIRED PASSWORD AND REENTER COMMAND
(IN) PASSWORD P
(IN) CATALOG DAT.GROUP,BASE=1,STATE=UPDATE
(IN) CATALOG DAT.GROUP(+1)
(IN) CATALOG DAT.GROUP(+2)
(IN) CATALOG DAT.GROUP(+3) _____ (07)
(OUT) % DMS05CC FILE NAME ALREADY CATALOGED OR INVALID OPERAND.
      COMMAND NOT PROCESSED
(IN) CATALOG DAT.GROUP(+3)
(IN) CATALOG DAT.GROUP(+5) _____ (08)
(OUT) % DMS06C7 INVALID GENERATION NUMBER IN COMMAND. NUMBERS MUST
      BE IN UNBROKEN ASCENDING ORDER. COMMAND NOT PROCESSED
(IN) CATALOG DAT.GROUP(+4)
(IN) CATALOG DAT.GROUP(+5)
(IN) FSTATUS DAT.GROUP,GEN=YES,CATALOG
(OUT) 00000000 :N:$USR123.DAT.GROUP (FGG)
      SHARE = NO ACCESS = WRITE
      ACL = NO AUDIT = NONE DESTROY = YES
      CRDATE = 1990-12-18 EXDATE = 1990-12-18 LADATE = NONE
      RDPASS = YES WRPASS = NONE EXPASS = NONE
      ACCESS# = 000 VERSION = 000
      LARGE = NO BACKUP = C MIGRATE = ALLOWED
00000000 :N:$USR123.DAT.GROUP(*0002)
      SHARE = NO ACCESS = WRITE
      ACL = NO AUDIT = NONE DESTROY = YES
      CRDATE = NONE EXDATE = NONE LADATE = NONE
      RDPASS = YES WRPASS = NONE EXPASS = NONE
      ACCESS# = 000 VERSION = 000
      LARGE = NO BACKUP = C MIGRATE = ALLOWED
00000000 :N:$USR123.DAT.GROUP(*0003)
      SHARE = NO ACCESS = WRITE
      ACL = NO AUDIT = NONE DESTROY = YES
      CRDATE = NONE EXDATE = NONE LADATE = NONE
      RDPASS = YES WRPASS = NONE EXPASS = NONE
      ACCESS# = 000 VERSION = 000
      LARGE = NO BACKUP = C MIGRATE = ALLOWED
(OUT) 00000000 :N:$USR123.DAT.GROUP(*0004)
      SHARE = NO ACCESS = WRITE
      ACL = NO AUDIT = NONE DESTROY = YES
      CRDATE = NONE EXDATE = NONE LADATE = NONE
      RDPASS = YES WRPASS = NONE EXPASS = NONE
      ACCESS# = 000 VERSION = 000
      LARGE = NO BACKUP = C MIGRATE = ALLOWED
00000000 :N:$USR123.DAT.GROUP(*0005)
      SHARE = NO ACCESS = WRITE
      ACL = NO AUDIT = NONE DESTROY = YES
      CRDATE = NONE EXDATE = NONE LADATE = NONE
      RDPASS = YES WRPASS = NONE EXPASS = NONE
      ACCESS# = 000 VERSION = 000
      LARGE = NO BACKUP = C MIGRATE = ALLOWED
(OUT) 00000000 :N:$USR123.DAT.GROUP(*0006)
      SHARE = NO ACCESS = WRITE
      ACL = NO AUDIT = NONE DESTROY = YES
      CRDATE = NONE EXDATE = NONE LADATE = NONE
      RDPASS = YES WRPASS = NONE EXPASS = NONE
      ACCESS# = 000 VERSION = 000
      LARGE = NO BACKUP = C MIGRATE = ALLOWED
      :N: PUBLIC: 6 FILES RES= 0 FREE= 0 REL= 0 PAGES

```

(01) The catalog entry for the file FILE.NEW is created. The file attributes

- (ACCESS=..., SHARE=..., RDPASS=..., ...) are set to their default values.
- (02) A read password is defined. The password is replaced by the letter "P" in the log.
  - (03) The catalog entry for the file generation group DAT.GROUP is created. Up to five generations can be cataloged with one CATALOG command. A read password is defined. BACKUP=C is specified for file saving; DESTROY=YES provides for data destruction during erasure.
  - (04) The first file generation is cataloged (specifying the absolute generation number).
  - (05) The second file generation is cataloged (specifying the relative generation number). An error message is returned because the base value has not yet been defined.
  - (06) The first file generation is defined as the base generation. The read password must be entered in the task password table (PASSWORD command) before the catalog entry is updated.
  - (07) An attempt is made to catalog the third file generation, again specifying the absolute generation number. An error message is returned.
  - (08) A catalog entry for a file generation cannot be created until the predecessor generation has been cataloged.

## **CATJV Catalog job variable**

Application group: Job variable functions (page 36)

The CATJV command is only available with the software product JV (see the "Job Variables" manual [11]).

### **Command description**

The CATJV command allows a job variable entry in the catalog to be created or updated.

### **Formats and operand descriptions**

Given below are the formats and operand descriptions for


- permanent job variables and
- temporary job variables.

## Permanent job variables

Operation	Operands
CATJV	<pre> jvname1 [, jvname2]  [, STATE={   NEW   UPDATE }] [, ACCESS={   WRITE   READ }]  [, SHARE={   NO   YES }] [, RETPD=days]  [, RDPASS={   NONE   password }] [, WRPASS={   NONE   password }]  [, MONJV={   UNCHANGED   NO }]  [, BASIC-ACL={   NONE   ( {     OWNER=access rights     GROUP=access rights     OTHERS=access rights   } [, { ... } ] ) } ] </pre>

- jvname1** Specifies the fully-qualified name under which the job variable is cataloged. Only the system administrator is allowed to specify the user ID of another user.
- jvname2** Declares a new name for the job variable "jvname1". User ID and catalog ID must not be specified, since this would mean a change of owner or catalog. This operand only takes effect when "STATE=UPDATE" has likewise been supplied.
- "jvname2" may be a temporary or permanent job variable. If "jvname1" is a permanent job variable, and "jvname2" is a temporary one, the protection attributes must be reset to default values at the same time.

- ACCESS** Defines possible access to the job variable.
- =READ** Specifies that the job variable may only be read.
  - =WRITE** Both read and write access are permitted for the job variable.
- BASIC-ACL** Specifies whether the job variable is to be protected by means of a BACL entry.
- =NONE** The job variable is not to be protected by a BACL entry.
  - =(...)** The job variable is to be protected by a BACL entry.
    - OWNER=...** Grants the access rights specified below to the owner.
    - GROUP=...** Grants the access rights specified below to the user's user group.
    - OTHERS=...** Grants the access rights specified below to the other user groups.
- access-rights
- mean:
- $$\left\{ \begin{array}{l} \text{NO-ACCESS} \\ \left( \left[ \text{READ}=\begin{cases} \text{NO} \\ \text{YES} \end{cases} \right] \left[ \text{WRITE}=\begin{cases} \text{NO} \\ \text{YES} \end{cases} \right] \right) \end{array} \right\}$$
- NO-ACCESS** No access authorization
  - READ** Read access: yes/no
  - WRITE** Write access: yes/no
- MONJV** Specifies whether or not the monitoring job variable should continue to be protected.
- =UNCHANGED** Specifies that the protection for the monitoring job variable is to remain unchanged.
  - =NO** Cancels the system area (bytes 1 through 128) protection of a previous monitoring job variable. This specification is only effective with **STATE=UPDATE** and, furthermore, only when the job to be monitored no longer exists.

RDPASS	Defines whether or not the job variable is to be protected by means of a read password. The read password must be an expression one to four characters in length. Character, hexadecimal and integer formats are permitted. A password such as X'00000000' is ignored.
=NONE	Indicates that the job variable is not to be protected by a read password, and that any previously defined read password is to be cancelled.
=password	Specifies the read password with which the job variable is to be protected.
RETPD=days	Specifies a retention period in days (0...32767), during which the job variable must not be updated or deleted (write protection). The job variable must already be cataloged; if STATE=NEW is specified at the same time, RETPD is ignored. Default value: days=0, i.e. the job variable may be updated immediately.
SHARE	Defines whether foreign user IDs are to be granted access to the job variable.
=YES	Specifies that the job variable may be accessed from a foreign user ID.
=NO	Specifies that the right to access the job variable is not to be granted to foreign user IDs. Only the owner or the system administrator may then access the job variable.
STATE	Specifies whether a new catalog entry is to be created for a job variable or an existing catalog entry is to be updated.
= <u>NEW</u>	A new catalog entry is to be created.
=UPDATE	An existing catalog entry is to be updated.
	 Only explicitly specified catalog fields are updated with STATE=UPDATE. Omitted operands are not set to default values.

WRPASS	Defines whether or not the job variable is to be protected by means of a write password. The write password must be an expression one to four characters in length. Character, hexadecimal and integer formats are permitted. A password such as X'00000000' is ignored.
=NONE	Indicates that the job variable is not to be protected by a write password, and that any previously defined write password is to be cancelled.
=password	Specifies the write password with which the job variable is to be protected.

For examples see the "Job Variables" manual [11].



## Temporary job variables

Operation	Operands
CATJV	#jvname1 [, jvname2]  [ , STATE={ <u>NEW</u> } { UPDATE } ]

- #jvname1** Name of the temporary job variable to be cataloged. "**#**" is the prefix identifying temporary files and job variables; it is defined by the system administrator by means of the system parameter **TEMPFILE**.
- jvname2** Declares a new name for the job variable "jvname1". User ID and catalog ID must not be specified, since this would mean a change of owner or catalog. This operand only takes effect if "STATE=UPDATE" has likewise been specified. "jvname2" may be a temporary or a permanent job variable.
- STATE** Specifies whether a new catalog entry is to be created or whether an existing one is to be updated.
- =NEW** Creates a new catalog entry under the name  
S.TMP.tsn.jvname1  
This is the default value.
- =UPDATE** Specifies that a temporary job variable is to be renamed.

The unmodifiable protection attributes for temporary job variables are normally set to the following default values:

RETPD=0 , RDPASS=NONE , WRPASS=NONE , SHARE=NO , ACCESS=WRITE

A temporary job variable appears in system messages and in the output of **SHOW-CJC-STATUS** under its internal name **S.TMP.tsn.jvname**.

Renaming can be used to convert a temporary job variable into a permanent one. When a permanent job variable is converted to a temporary one, protection attributes must be reset to default values at the same time.

For examples see the "Job Variables" manual [11].

## CHANGE      Change TFT entry

Application group: Device and volume reservation (page 29)

### Command description

The CHANGE command can be used to change the file link name in a task file table (TFT) entry. All other values in this TFT entry remain unaltered. The command cannot be applied to the TFT entry for an open file.

### Format and operand description

Operation	Operands
CHANGE	[link1], link2

- link1                      Specifies the file link name (1 to 8 bytes).  
If this entry is omitted, the first TFT entry with the link name C'\_\_\_\_\_' is processed.
- link2                      Specifies a new file link name which is to replace the previous link name.

**Example:**

```

(IN)      FILE SAL.FILE1, LINK=SORTIN _____ (01)
(IN)      RDTFT _____ (02)
(OUT)    %
          ...
          ...
          %      LINK=SORTIN
          FILE=:N:$USR123.SAL.FILE1
          %
          ...
          ...
(IN)      CHANGE SORTIN, SORTOUT _____ (03)
(IN)      RDTFT _____ (02)
(OUT)    %
          ...
          ...
          %      LINK=SORTOUT
          FILE=:N:$USR123.SAL.FILE1
          %
          ...
          ...
(IN)      RELEASE SORTOUT _____ (04)

```

(01) Assigns the link name SORTIN to the file SAL.FILE1.

(02) Reads the task file table.

(03) Changes the link name SORTIN to SORTOUT.

(04) The link name SORTOUT is to be deleted.

For a further example, see the HOLD command.

## CONNECTION      Enter virtual connection in CLT

Application group: Transaction mode (DCAM) (page 33)

### Command description

The CONNECTION command enables DCAM users to store or delete data concerning a virtual connection in a process-specific table known as the Communication Link Table (CLT).

The values contained in this table supplement or replace the relevant entries in the connection control block CCB (Assembler) or in the connection structure (COBOL) while such a connection is being established. The linkage between the CLT entry and this program area is provided by the link name, which must be specified both in the command and in the program. (See also the "DCAM Program Interfaces" [6] and "DCAM Macros" [17] manuals.)

The CONNECTION command can only affect connection setup if a link name has been defined in the program.

### Format and operand description

Operation	Operands
<pre>{CONNECTION} {CONN      }</pre>	<pre>[partnername] , LINK=linkname [ , PRONAME=procname] [ , USERFLD=userfield] [ , LOGPW=password]</pre>

- partername** Specifies the name of the communication partner with whom a connection is to be established.
- The name, which may be up to 8 bytes long, must consist of EBCDIC characters of which the first must be a letter, "@", "#", or "\$".
- LINK=linkname** Specifies a link name consisting of no more than 8 EBCDIC characters, of which the first must be a letter, "@", "#", or "\$".
- This link name establishes the link between the entry in a process-specific table (CLT) and an area in the program (CCB or connection structure).
- Special function: If LINK is the only operand in the CONNECTION command, the CLT entry associated with the specified link name is deleted.
- LOGPW=password** Specifies the password which the communication partner must give when he wishes to establish a connection.
- The password is 4 bytes long and can be a character constant (C'xxxx') or hexadecimal constant (X'xxxxxxxx').
- PRONAME=procname**
- Specifies the name of the processor node in which the partner is located as a station.
- This name consists of no more than 8 EBCDIC characters, of which the first must be a letter, "@", "#", or "\$".
- USERFLD=userfield**
- Specifies a field where you can store additional information which serves as a user-own identifier for the virtual connection.
- "userfield" is 4 bytes long and can be a character constant (C'xxxx') or hexadecimal constant (X'xxxxxxxx').
- This operand is ignored in DCAM-COBOL programs.

### Example

The CONNECTION command shown below has the following effects during connection setup:

1. The name of the processor node is changed and
2. the password FR is specified.

Fig. CONN-1 Entering the virtual connection in the CLT

## COPY Copy file

Application group: File processing (page 26 ff.)

### Command description

The COPY command can be used to copy files, file generations or file generation groups

- from disk to disk
- from disk to tape
- from tape to disk.

The files are copied block-by-block; thus the command cannot be used to change the file attributes (exception: block control format, see below).

If the target file has not yet been cataloged, it is automatically created on public disk. If it is to be created on private disk or tape, it must be created with the FILE command (DEVICE and VOLUME operands) prior to copying.

The primary and secondary allocations for a target file automatically created on public disk are taken from the source file if the latter is also on disk. If the source file is on tape, default values are used for the primary and secondary allocations.

The primary and secondary allocations for a cataloged target file on disk are only changed if the target file is smaller than the source file and the secondary allocation is  $\neq 0$ .

The COPY command is rejected if

- write access is not allowed for the target file (ACCESS=READ, EXDATE > current date, file protected by write password or BASIC-ACL);
- the secondary allocation is 0 and the primary allocation is too small to accommodate the file to be copied;
- the source file is an ISAM file on private disks with separate index and data sections.

If a file on private disk has only one entry in the system catalog but none in the F1 label, the system catalog entry will be deleted. If this file is the source file, the command will be rejected after the catalog entry has been deleted. If it is the target file, a new file will be created in public storage.

COPY operates internally with the link name DMCOPY11 for source files and DMCOPY22 for target files.

BACL-protected PLAM libraries cannot be copied.

## Changing the block control format


In addition to the keyed file format (BLKCTRL=PAMKEY), BS2000 versions as of V10 also provide a non-keyed file format (BLKCTRL=DATA/NO). The PAMCONV utility routine is normally used to convert files from one format to the other. However, in certain cases it is also possible to use COPY to generate a target file whose block control format is different from that of the source file. In order to achieve this, the command `FILE target-file,link=dmcopy22,BLKCTRL=...` must be issued before the COPY command. The COPY command must then contain the operand specification `BLKCTRL=IGNORE/CHECK`.

## Copying tape files

Source files on tape with `BLKCTRL=PAMKEY` must have standard block length. Target files with `BLKCTRL=PAMKEY` are created on tape with standard block length. (Standard block length for tape files: 16 bytes (PAM key) + 2048 bytes (PAM block)).


Tape files with `BLKCTRL=DATA/NO` whose `BLKSIZE` is a multiple of 2048 bytes can also be copied with COPY.

If NK files are copied to tape, the `BLKCTRL` information is lost when the catalog entry is deleted. If the file is to be copied back again, the COPY command must be preceded by a FILE command with the operands `STATE=FOREIGN` and `LINK`. The correct value for the `BLKCTRL` operand must be specified in the FILE command, i.e. either `NO` or `DATA`, depending on the file format.

 If a K file (`BLKCTRL=PAMKEY`) is inadvertently copied to an NK file (`BLKCTRL=DATA`), the resulting disk file cannot be read, because the first 16 bytes of each PAM page, which contain data when `BLKCTRL=PAMKEY` applies, are overwritten with management information.

COPY allows only one tape to be used for source and target file (no continuation tape). Although a number of files can be copied to one tape, it is not possible to copy files spread over a number of tapes.

Before copying a tape file that has not yet been cataloged (remote tape file), the file attributes must be defined by issuing a FILE command with the file link name valid for the COPY command:

 On completion of the copying procedure, file link names `DMCOPY11` and `DMCOPY22` are implicitly released. Under certain circumstances, this can cause the release of an existing TFT entry.

```
/FILE pathname1, LINK=DMCOPY11, STATE=FOREIGN, BLKCTRL=...
```



## Copying file generation groups

Only a file generation group comprising SAM file generations with the same attributes (e.g. same record and block lengths, same record format, same BLKCTRL value) can be copied to a single file or file generation. This file generation must not belong to the file generation group to be copied.

A file generation group can only be copied to another file generation group if one of the following conditions is satisfied:

- The two file generation groups have matching group entries (i.e. identical values for GEN, FIRST, LASTGN and BASE).  
The file generation group to which the copy is to be written must already contain the FIRST to LASTGN generations (i.e. the generations must be cataloged).
- The GEN value is identical for both file generation groups and the generation group to which the copy is to be written still contains no generation (i.e. FIRST, LASTGN and BASE have the value zero).

## Remote file access (see also "RFA" manual [12])

If a file is copied from remote system to remote system with input/output on two different systems, then this is supported by the higher-ranking execution routine. The local system serves as an intermediate station for the data transfer. The RFASTART command must be issued for both remote systems before the copying process commences.

When a remote file is copied to a local file with the SAME operand, the passwords are not included.

If NK files are to be transferred to systems with an operating system version earlier than BS2000 V10 or if a K file from a system with an operating system version earlier than BS2000 V10 must be stored on an NK disk,

- the COPY command must be issued in the system with BS2000 V10;
- the file must be converted by means of the PAMCONV utility routine before or after copying (see "Utility Routines" manual [16]).

If operands which are invalid in the remote system's BS2000 version are specified, the entire command is rejected (BLKCTRL=CHECK/IGNORE for BS2000 < V10).

An NK file can be copied from BS2000 V10.0A to BS2000 V9.5 or an earlier version. However, if the file has a data format which is not supported in the target system, this may cause problems.

**Format and operand description**

Operation	Operand
COPY	<p>pathname1,pathname2 [ , SAME ] [ , WRITE= { <u>REPLACE</u> NEW [ , DIALOG= { <u>YES</u> NO } ] } ]</p> <p>[ , BLKCTRL= { <u>IGNORE</u> CHECK } ]</p>

pathname1	<p>stands for: [:catid:][\$userid.]filename</p> <p>"pathname1" is the file/file generation/file generation group to be copied. The file/file generation/file generation group must be shareable (SHARE=YES; BASIC-ACL with access right) if it is not cataloged under the ID of the user issuing the job.</p> <p>If "pathname1" is an FGG, "pathname2" must also be an FGG unless the FGG "pathname1" consists of SAM file generations with identical attributes as regards record format, record length and block size. In this case, it is possible to copy into a single file or into a file generation, provided that this file generation does not belong to the FGG which is to be copied.</p>
catid	<p>Catalog ID of the pubset on which the file is stored. Default value: catalog ID assigned to the user ID (JOIN entry).</p>
userid	<p>User ID to which the file is assigned. Default value: User ID from the LOGON command.</p>
filename	<p>Fully-qualified name of the file, file generation or file generation group which is to be copied.</p>
pathname2	<p>stands for: [:catid:][\$userid.]filename</p> <p>"pathname1" and "pathname2" must not designate the same file. "pathname2" specifies the target file.</p> <p>If "pathname2" is not yet cataloged, only the user ID entered in the LOGON command may be specified. If "pathname2" is already cataloged, write access must be permitted, i.e. any existing write password must be specified, there must be no retention period specified for the file, the access type must be ACCESS=WRITE and the file must have BASIC-ACL access rights.</p>

If "pathname2" is cataloged under a foreign user ID, this user ID must be specified and the file must be shareable (SHARE=YES in the catalog entry and BASIC-ACL access rights).

If "pathname2" is an FGG, "pathname1" must also be an FGG.

catid	Catalog ID of the pubset on which the file is stored. Default value: catalog ID assigned to the user ID (JOIN entry).
userid	User ID to which the file is assigned. Default value: user ID from the LOGON command.
filename	Fully-qualified name of the copy (target file).
BLKCTRL	Specifies whether the block control attributes (or the TFT entry for the file link name DMCOPY22) of the target file and source file may differ. Default value: The target file or TFT entry must have the same block control attributes as the source file.
=IGNORE	An inconsistency in block control attributes may be ignored in the following cases:

Source file	Target file
PAMKEY	DATA (for disk file only)
PAMKEY	NO
DATA (for disk file only)	PAMKEY
NO	PAMKEY



If a file with BLKCTRL=PAMKEY is copied to a file with BLKCTRL=DATA/NO, data in the user section of the PAM key is lost.

=CHECK	Inconsistencies in block control attributes are to be ignored only if no user data in the user section of the PAM key would be lost in the transfer. If the user section of the PAM key contains no user information, inconsistencies in block control attributes can be ignored in the following cases:
--------	--

Source file	Target file
PAMKEY	DATA (for disk files only)
PAMKEY	NO

- DIALOG**            The DIALOG operand is only evaluated in interactive mode if WRITE=NEW is specified at the same time.
- =YES            If the file "pathname2" contains data, i.e. the target file is not empty, the operating system should ask if the file is to be overwritten before copying the source file. If "pathname2" is empty or is a tape file, the operating system does not ask before overwriting it. DIALOG=YES is the default setting.
- =NO             If "pathname2" already exists, the COPY command is rejected; "pathname2" is not overwritten.
- 
- SAME**             The "pathname2" copy receives the same file backup and file protection attributes as "pathname1" (the values for BACKUP, LARGE, RETPD, DESTROY, SHARE, ACCESS, MIGRATE and the same passwords). The AUDIT value, however, is not transferred. If "pathname2" is not yet cataloged, and if SAME is not specified, the new file is created with the default attributes (see the default settings for the operands of the CATALOG command, e.g. SHARE=NO, ACCESS=WRITE etc.).
- The operand SAME is ignored if "pathname2" is cataloged under a foreign user ID or is a file generation (its file attributes are then defined in the group entry).
- If a temporary file is copied into a permanent file with the operand SAME specified, the permanent file receives the attribute BACKUP=E, i.e. the new file is ignored for ARCHIVE save runs. If the new file is to be saved automatically using ARCHIVE, the BACKUP value must be changed by means of a CATALOG command.
- If "pathname1" is protected by a BASIC-ACL entry or an ACL entry, the following applies if SAME is specified:
- If the target file is created on a private disk, it receives the same BASIC-ACL protection attributes as the source file. If an ACL entry exists for the source file, the default protection attributes - SHARE=NO and ACCESS=WRITE - are assigned to the target file.
  - If the target file is created on a magnetic tape, the default protection attributes - SHARE=YES and ACCESS=WRITE - are assigned to it, regardless of the protection attributes defined for the source file in the ACL or BASIC-ACL entry.

- If the source file is not cataloged under the user ID from which COPY is invoked, the default protection attributes are assigned to the target file, regardless of the protection attributes assigned to the source file in the ACL or BASIC-ACL entry. The default protection attributes are: SHARE=NO and ACCESS=WRITE for a disk file SHARE=YES and ACCESS=WRITE for a tape file

**WRITE** Specifies whether an existing target file "pathname2" is to be overwritten. If "pathname2" is a tape file or is empty, the WRITE operand will be ignored and the old file "pathname2" overwritten without warning.

**=REPLACE** The file/file generation/file generation group designated by "pathname2" is overwritten without output of a message; default value.

**=NEW** "pathname2" is not overwritten (see also DIALOG operand)

## Examples

### Example 1: SAME operand

```
(IN)  FS MAX.FILE.3
(OUT) % DMS0533 REQUESTED FILE NOT CATALOGED ON PUBSET 'N'. COMMAND TERMINATED
(IN)  COPY MAX.DAT,MAX.FILE.3,SAME
(IN)  FS MAX.FILE.3,A
(OUT) 00000003 :N:$USR123.MAX.FILE.3
      FCBTYP = SAM          VSNTYPE = PUB
      LASTPG = 00000001     2ND ALLO= 00003
      SHARE  = NO          ACCESS  = WRITE
      ACL    = NO          AUDIT   = NONE          DESTROY = NO
      CRDATE = 1990-12-18  EXDATE  = 1990-12-18      LADATE  = 1990-12-18
      RDPASS = YES          WRPASS  = NONE          EXPASS  = NONE
      ACCESS# = 001        VERSION = 001
      LARGE  = NO          BACKUP  = A           MIGRATE = ALLOWED
      BLKTYPE = STD        BLKSIZE = 002048      BLKCTRL = PAMKEY
      RECFORM = (V,N)      RECSIZE = 000000
      VSN/DEV/EXT =        PUBN01 / D3480      / 001
      EXTCNT = 1
      :N: PUBLIC: 1 FILE RES= 3 FREE= 2 REL= 0 PAGES
```

## Example 2: Copying temporary files

```

(IN) COPY MAX.FILE,#TEMP
(IN) FSTAT #TEMP,ALL
(OUT) 00000003 :N:$USR123.S.212.0X5S.TEMP
      FCBTYPE = SAM          VSNTYPE = PUB
      LASTPG = 00000001     2ND ALLO= 00003
      SHARE = NO           ACCESS = WRITE
      ACL = NO             AUDIT = NONE          DESTROY = NO
      CRDATE = 1990-12-18  EXDATE = 1990-12-18    LADATE = 1990-12-18
      RDPASS = NONE        WRPASS = NONE        EXPASS = NONE
      ACCESS# = 001        VERSION = 001
      LARGE = NO           BACKUP = E
      BLKTYPE = STD        BLKSIZE = 002048     MIGRATE = INHIBIT
      RECFORM = (V,N)      RECSIZE = 000000     BLKCTRL = PAMKEY
      VSN/DEV/EXT = PUBN00 / D3480 / 001
      EXTCNT = 1
:N: PUBLIC: 1 FILE RES= 3 FREE= 2 REL= 0 PAGES

(IN) COPY #TEMP,MAX.PERM.1
(IN) COPY #TEMP,MAX.PERM.2,SAME
(OUT) FSTAT MAX.PERM.,ALL
(OUT) 00000003 :N:$USR123.MAX.PERM.1
      FCBTYPE = SAM          VSNTYPE = PUB
      LASTPG = 00000001     2ND ALLO= 00003
      SHARE = NO           ACCESS = WRITE
      ACL = NO             AUDIT = NONE          DESTROY = NO
      CRDATE = 1990-12-18  EXDATE = 1990-12-18    LADATE = 1990-12-18
      RDPASS = NONE        WRPASS = NONE        EXPASS = NONE
      ACCESS# = 001        VERSION = 001
      LARGE = NO           BACKUP = A
      BLKTYPE = STD        BLKSIZE = 002048     MIGRATE = ALLOWED
      RECFORM = (V,N)      RECSIZE = 000000     BLKCTRL = PAMKEY
      VSN/DEV/EXT = PUBN01 / D3480 / 001
      EXTCNT = 1
00000003 :N:$USR123.MAX.PERM.2
      FCBTYPE = SAM          VSNTYPE = PUB
      LASTPG = 00000001     2ND ALLO= 00003
      SHARE = NO           ACCESS = WRITE
      ACL = NO             AUDIT = NONE          DESTROY = NO
      CRDATE = 1990-12-18  EXDATE = 1990-12-18    LADATE = 1990-12-18
      RDPASS = NONE        WRPASS = NONE        EXPASS = NONE
      ACCESS# = 001        VERSION = 001
      LARGE = NO           BACKUP = E
      BLKTYPE = STD        BLKSIZE = 002048     MIGRATE = INHIBIT
      RECFORM = (V,N)      RECSIZE = 000000     BLKCTRL = PAMKEY
      VSN/DEV/EXT = PUBN02 / D3480 / 001
      EXTCNT = 1
(OUT) :N: PUBLIC: 2 FILES RES= 6 FREE= 4 REL= 0 PAGES

```

*Example 3: Overwriting files with/without dialog*

```
(IN) COPY MAX.FILE,MAX.FILE.3,WRITE=NEW
(OUT) % DMS0518 FILE ':W:$US123456.MAX.FILE.3' ALREADY EXISTS. OVERWRITE ?
      REPLY (Y=YES; N=NO)?
(IN) Y
(IN) COPY MAX.FILE,MAX.FILE.3,WRITE=NEW
(OUT) % DMS0518 FILE ':W:$US123456.MAX.FILE.3' ALREADY EXISTS. OVERWRITE ?
      REPLY (Y=YES; N=NO)?
(IN) N
(OUT) % DMS0519 COPY COMMAND WITHDRAWN BY CALLER
```

*Example 4: Copying files to private disk*

```
(IN) FILE MAX.PRIV.DAT.2,DEVICE=D3480,VOLUME=WORK01 _____ (01)
(IN) COPY MAX.FILE,MAX.PRIV.DAT.2 _____ (02)
(IN) FSTAT MAX.PRIV.DAT.2,ALL _____ (03)
(OUT) 00000003*:W:$US123456.MAX.PRIV.DAT.2
      FCBTYPE = SAM      VSNTYPE = PVT      LASTPG = 0000001      2ND ALLO= 00009
      SHARE = NO        ACCESS = WRITE
      ACCESS# = 001     CRDATE = 88-11-10   EXDATE = 88-11-10   LADATE = 88-11-10
      RDPASS = NONE     WRPASS = NONE     EXPASS = NONE
      VERSION = 001    LARGE = NO        BACKUP = A          MIGRATE = ALLOWED
      DESTROY = NO     AUDIT = NONE
      BLKTYPE = STD    BLKSIZE = 002048   RECFORM = (V,N)    RECSIZE = 00000
      BLKCTRL = PAMKEY
      VSN/DEV/EXT =    WORK01 / D3480 / 001
      EXTCNT = 1
      :W: PRIVATE: 1 FILE. RES= 3, FREE= 2, REL= 0 PAGES
```

- (01) The FILE command creates the catalog entry for a file, and simultaneously requests the specified private disk.
- (02) The file "copy" is stored on the private disk specified in the FILE command.
- (03) This is the catalog entry of the file "copy" stored on private disk. Note the fields VSNTYPE=PVT, VSN/DEV/EXT=WORK01/D3480/001, which indicate the disk on which the file has been stored.

## CREATE-ISAM-POOL      Create or join ISAM pool

Application group: File processing (page 26 ff.)

### Command description

The CREATE-ISAM-POOL command can be used to either create a new ISAM pool or to join in an existing one. An ISAM pool is identified by

- its name,
- its catalog ID,
- its scope,
- the type of buffering, and
- its size.

If a new ISAM pool is to be opened, the name must be explicitly specified; the scope (task, system), size, and catalog ID are optional.

### Format and operand description

Operation	Operands
CREATE-ISAM-POOL	POOL-NAME=name  [, CAT-ID={ *DEFAULT-PVS catid }]  [, SCOPE={ TASK (WRITE-IMMEDIATE={ NO YES }) HOST-SYSTEM ( [WRITE-IMMEDIATE={ NO YES }] [, CREATION-MODE={ ANY NEW }) }]  [, SIZE={ STD size }] 


**POOL-NAME**      Specifies the name of the ISAM pool.

=name

Name of the ISAM pool, 1 to 8 characters in length.

The first character must be a letter or one of the special characters #, @; the remaining characters may consist of any combination from the character set (A,...,Z,0,...,9,\$,#,@).



CAT-ID	Specifies the PVS catalog ID of the host system where the ISAM pool exists or is to be created. Each host system is assigned a unique catalog ID. (Different catalog IDs refer to different ISAM pools.)
<u>=*DEFAULT-PVS</u>	The catalog ID assigned to the user ID (JOIN entry) is used; this is the default value.
=catid	Catalog ID of a pubset (PVS).
SCOPE	Defines the scope (participants) of the ISAM pool.
<u>=TASK(...)</u>	The ISAM pool is available only to the user's own task; default value.
WRITE-IMMEDIATE	
<u>=NO</u>	Modified blocks of a file are not immediately saved to disk unless WROUT=YES was specified earlier in the FILE command or in the FCB; default value.
=YES	Modified blocks of the processed file are saved to disk immediately.
<u>=HOST-SYSTEM(...)</u>	The ISAM pool is available to all tasks.
WRITE-IMMEDIATE	
=NO	Modified blocks of a file are not immediately saved to disk.
	 The request to open the file will be rejected if, on the one hand, the operand WRITE-IMMEDIATE=NO is specified for HOST-SYSTEM, but on the other, WROUT=YES has been entered in the TFT or the FCB of the file to be processed.
=YES	Modified blocks of the processed file are immediately saved to disk; default value.
CREATION-MODE	
	Specifies, for a task-wide ISAM pool, whether it is essential that a new pool be created, or whether the task may be added to any existing ISAM pool with the same pool name, catalog ID, and scope.
<u>ANY</u>	If the named ISAM pool already exists, the task is added to this pool; otherwise, an ISAM pool of the specified size is created. "ANY" is the default value.
NEW	A new ISAM pool is to be created. This command is rejected if an ISAM pool with the same name, catalog ID, and scope already exists on the host system.

- SIZE** Specifies the size of the ISAM pool in PAM pages. Since pools are created in units of segments (1 Mb for XS systems, 64 Kb for others), the specified value may be internally rounded up to the next segment boundary.
- =STD** If a new ISAM pool is to be created, it should be the standard size, which is defined when the system is generated. If the new pool is to be added to an existing ISAM pool, the size of the existing pool is retained; default value.
- =size** Size of the ISAM pool in PAM pages:  
 $32 \leq \text{size} \leq 32767$  for XS systems with 31-bit addressing;  
 $32 \leq \text{size} \leq 2048$  for all other systems, where the maximum size of the user address space defined by the system administrator represents the upper limit.  
Use of the minimum size of 32 PAM pages restricts processing to files with a maximum block size of (STD,6) (cf. the FILE command).  
When an existing ISAM pool is to be joined, the SIZE specification is ignored.

## DATA Open spoolin file

Application group: File processing (page 26 ff.)

### Command description

The DATA command catalogs and opens a file during spoolin from floppy disk.

Records following the DATA command are entered as data records in the opened file (SPOOLIN file). This file is closed if an END command is encountered, in which case spoolin branches back to the SPOOLIN file.

A single job can contain multiple DATA commands, each of which closes the previously opened SPOOLIN file. In this case, the END commands between the DATA commands can be omitted.

The command will be rejected in interactive mode in an ENTER job.

If a spoolin operation aborts, the DATA file continues to exist. This also applies if the LOGON command is rejected.

The first DATA command must be issued immediately after the LOGON command for the batch job.

The data records to be entered in a SPOOLIN file may also be commands. However, the LOGON or PROCEDURE command must immediately follow the DATA command.

ISAM records start with an 8-digit key field, the increment being 100; i.e. the first record has the key 00000100, the second 00000200, etc.

A file to be created on private disk must already have been cataloged before the batch job is started.

After a file has been closed, the following message appears in the printer listing for the job:

```
/* REMARK          n RECORDS SPOOLED IN DATA FILE
```

## Format and operand description

Operation	Operands
DATA	<pre> pathname [ ,FCBTYPE={ <u>SAM</u> } ] [ ,SPACE={ primary               (primary,secondary) } ]  [ { ,RECFORM=<u>V</u>[ ,STRIP={ YES }                           NO } ]   [ ,RECFORM=F [ ,RECSIZE=reclength ] ] </pre>

pathname      [:catid:][\$userid.]filename

catid          Catalog ID of the pubset on which the file is to be initialized.  
Default value: the catalog ID assigned to user ID (JOIN entry).



The specified pubset must have been imported when the DATA command is executed; otherwise, the DATA command will be rejected.

userid         User ID of the caller; may be omitted since the user ID from the LOGON command is the default value.

filename       Fully-qualified file name.  
The file is cataloged under this name. "filename" can also be the name of a temporary file (see the manual "DMS Introductory Guide and Command Interface" [8]).

Any existing file of the same name will be overwritten.  
Entering the name of a file generation or file generation group is not permitted; this also applies to the name of a file group in the case of disk files.

FCBTYPE

=SAM  
=ISAM

Specifies the access method for the file.

SPACE	"primary" specifies how much storage space in the file is to be newly allocated.
=primary	
=(primary,secondary)	"secondary" specifies how much storage space in the file is to be allocated automatically when additional storage space is required in the course of file processing (see also the FILE command, SPACE operand). If the operand is omitted, "primary" and "secondary" assume the values defined at system generation time (see the "System Installation" manual [13]).
RECFORM	Specifies variable-length records; i.e. every record has at its beginning the 4-byte record length field, in the first two bytes of which the record length is entered.
= <u>V</u>	
=F	Specifies fixed-length records.
STRIP	Removes blanks at the end of variable-length records (RECFORM=V).
=YES	Records containing only blanks are reduced to 1 blank.
= <u>NO</u>	Blanks at the end of a record are not removed.
RECSIZE	Specifies the record length for RECFORM=F.
=reclength	"reclength" is defined as follows:  For SAM files: $1 \leq \text{"reclength"} \leq 128$ For ISAM files: $1 \leq \text{"reclength"} \leq 136$  If the specified record length is greater than the length in the HDR1 label, the end of the entered record is filled with blanks. If the specified record length is smaller, the record is truncated. This operand, when omitted, always defaults to the maximum value.

**Error messages**

In the event of errors, the following messages are included in the SYSLST log. A branch is made to the next STEP command or to the LOGOFF command in each case.

- /\* ERROR      SEQUENCE ERROR IN DATA FILE**  
Error in command sequence (see above)
- /\* ERROR      FORMAT ERROR IN DATA FILE**  
Invalid file name (cf. fully-qualified file name)
- /\* ERROR      RESOURCE ERROR IN DATA FILE**  
No more memory space available (SPACE operand)
- /\* ERROR      FILETYPE ERROR IN DATA FILE**  
Contradictory file attributes (cf. FCBTYPE, RECFORM and RECSIZE operands)
- /\* ERROR      I/O ERROR IN DATA FILE**  
Error when accessing the DATA file

## DCLJV Define JV link name

Application group: Job variable functions (page 36)

The DCLJV command is only available with the software product JV (see also the "Job Variables" manual [11]).

### Command description

The DCLJV command is used to assign a link name to a job variable. It also creates a catalog entry if the JV did not already exist and if the user has catalog authorization (see CATJV).

The JV link names assigned to the JV names are listed in the JV Link Table.

Only one entry may exist in the JV Link Table for each link name. However, a job variable can have more than one link name.

### Format and operand description

Operation	Operands
DCLJV	jvname [, LINK=*jvlink]

jvname	Fully-qualified name of a permanent or temporary job variable. If the JV already exists, you must have access authorization. If it is a new job variable, it is cataloged under the specified name with the default functions of the CATJV command. In this case, only the system administrator may specify a user ID other than his own.
LINK=*jvlink	Indicates the link name (first character "**") under which the job variable can immediately be addressed within the job. The maximum length of the link name is 8 characters, including "**". If in a DCLJV command the same link name is specified as in a previous DCLJV command, the previous assignment is superseded by the new assignment.

For examples, refer to the "Job Variables" manual [11].

## DELETE-ISAM-POOL Delete ISAM pool/sever link to ISAM pool

Application group: File processing (page 26 ff.)

### Command description

The DELETE-ISAM-POOL command enables the user of one or more ISAM pools to clear down the connection to one or all of the pool(s). The specified ISAM pool is deleted if the caller is the last or only user of that pool. The command is rejected as long as there are pool link names for the caller.

### Format and operand description

Operation	Operands
DELETE-ISAM-POOL	POOL-NAME= { <ul style="list-style-type: none"> <li>*ALL</li> <li>name ( [CAT-ID=*DEFAULT-PVS/catid]               <ul style="list-style-type: none"> <li>[ , SCOPE=TASK/HOST-SYSTEM ] )</li> </ul> </li> </ul>

<b>POOL-NAME</b>	Indicates which connections to ISAM pool are to be cleared down.
= <b>*ALL</b>	Connections to all ISAM pools known to the task are to be cleared down.
= <b>name(...)</b>	The connection to the specified ISAM pool is cleared down.
<b>CAT-ID</b>	Specifies the catalog ID of the ISAM pool.
= <b>*DEFAULT-PVS</b>	The catalog ID assigned to the user ID is used (JOIN entry); default value.
= <b>catid</b>	Catalog ID of the pubset.
<b>SCOPE</b>	Indicates the scope of the command.
= <b>TASK</b>	The command only affects local ISAM pools; default value.
= <b>HOST-SYSTEM</b>	The command only affects global ISAM pools.



## DELON Delete ON command

Application group: Job variable functions (page 36)

The DELON command is only available with the JV software product (see "Job Variables" manual [11]).

### Command description

The DELON command deletes one or more ON commands.

### Format and operand description

Operation	Operands
DELON	$\left\{ \begin{array}{l} .name \\ onid \\ ALL \end{array} \right\}$

name	Label which identifies the ON command. If you have assigned the same label to more than one ON command, all effective ON commands identified by "name" are deleted.
onid	ON identifier which is output to SYSOUT; identifies the ON command to be deleted. "onid" is generated during the processing of a syntactically correct ON command and is a job-related local number. Thus, in interactive mode, ON commands which have no "name" identifier can also be deleted.
ALL	All effective ON commands defined within the user job are deleted.

For examples refer to the "Job Variables" manual [11].

## DO Initiate procedure file

Application group: Job control (page 22 ff.)

### Command description

The DO command is used to initiate a command sequence stored in a file (procedure file).

The procedure file is opened and the command sequence is executed. During execution, the symbolic operands contained in the command sequence are replaced by the current operand values specified in the command call. This operation is referred to as a DO procedure.

Procedure files are SAM or ISAM files with variable-length records (RECFORM=V). A procedure file is stored:

- as a cataloged (also temporary) file
- as a J-type member in a program library (LMS library routine)

The first data record in a procedure file is the PROCEDURE command. Subsequent records are commands or data. The last record is the ENDP command (see Figs. DO-1 and DO-2 on the next page).

When the DO command is issued, the procedure file is assigned to the (system) file SYSCMD. If data is also to be read from the procedure file, the system files SYSDTA and SYSCMD in the procedure file must be equated.  
[Command: /SYSFILE SYSDTA=(SYSCMD)].

Command procedures can be nested as often as required.

Nesting means:

Processing of procedure file commands is interrupted by a call to another procedure file. Once the ENDP command has been reached, the remaining commands in the most recently exited procedure file are read, etc.

DO procedures only result in "pseudo-nesting" (see Fig. DO-2), however, since once the ENDP command has been reached, the command following the first DO command is read. Genuine nesting can only be achieved by means of CALL procedures.

Fig. DO-1 DO procedure

Fig. DO-2 Nesting of DO procedures (pseudo nesting)

**Format**

Operation	Operands
DO	pathname [ , {sympar=[paramvalue]} , ... ] [sympar=]paramvalue

**Operand descriptions** see CALL command.

## DROP Cancel HOLD status

Application group: Device and volume reservation (page 29)

The DROP command is used to cancel the HOLD status imposed by the HOLD command on an entry in the task file table (TFT). If a RELEASE command is still pending for this entry, it will now be processed.

### Format and operand description

Operation	Operands
DROP	[link]

link Specifies the file link name, i.e. the name of the task file table entry for which the HOLD status is cancelled.  
If this entry is omitted, the first TFT entry with the file link name C'\_\_\_\_\_' is processed.

For examples, see the HOLD command.

## END Close spoolin file

Application group: File processing (page 26 ff.)

### Command description

The END command closes a file that was cataloged and opened by a DATA command. The command refers to job input from the batch terminal or from floppy disk and is processed during spoolin of the batch job.

An END command preceding a DATA command may be omitted since previously opened SPOOLIN files are automatically closed when a new DATA command is processed.

### Format

Operation	Operands
END	

# ENDON      Terminate ON statement sequence

Application group: Job variable functions (page 36)

The ENDON command is only available with the software product JV (see also the "Job Variables" manual [11]).

**Command description**

The ENDON command is used to terminate an ON or TIMEOUT statement sequence.

If no corresponding ON command can be found for an ENDON command (the command sequence contains only ENDON), then an appropriate SYSOUT message is output and processing continues with the next command.

If the ENDON command is missing from an ON/ENDON construct, the job will terminate abnormally in batch mode. In interactive mode, the procedure is terminated.

**Format**

Operation	Operands
ENDON	

Examples can be found in the "Job Variables" manual [11].

## ENDP Terminate procedure file

Application group: Job control (page 22 ff.)

### Command description

The ENDP command is used to terminate a procedure. If the procedure was called with a DO command, ENDP moves the SYSCMD system file to the primary command input (terminal in interactive mode; SPOOLIN or ENTER file in batch mode), or to the last procedure level exited with CALL. If the procedure was called with the CALL command, SYSCMD returns to the calling procedure (see Fig. ENDP-1).

Following the ENDP command, the SYSDTA, SYSIPT, SYSLST, SYSOPT, SYSOUT and TASKLIB assignments are restored as they were before the procedure level was called. Any file assigned in the procedure by SYSDTA SYSLST=... will be closed when ENDP is issued.

If the ENDP command is issued in ESCAPE mode, SYSCMD returns to the previous procedure level (cf. RTI command: procedure execution is resumed from the point it was interrupted with ESCAPE).

If ENDP and DO commands are specified in ESCAPE mode, SYSCMD will be returned to the preceding procedure level, unless SYSCMD is already at primary command input level (level 0).

After the ENDP command or execution of the DO procedure, a message is output with the updated level number of the nested procedure.

*Exception:* No message is issued when the primary command input level has been reached.



Fig. ENDP-1 Combination of DO and CALL procedures

The arrows show the route of the system file SYSCMD, i.e. command input.

### Format

Operation	Operands
ENDP	

For examples see the PROCEDURE, ESCAPE and RTI commands.

## ENTER Process ENTER job

Application group: Job control (page 22 ff.)

### Command description

The ENTER command is used to transfer a batch job stored in an (ENTER) file to the operating system for processing.

The (ENTER) file is a cataloged file or a library member. The ENTER command can be issued in both command mode and program mode (see also "Executive Macros" manual [5]).

The new job is given its own task sequence number (TSN) and is executed in a separate task, regardless of which task called it. The specifications in the ENTER command indicate the (ENTER) file, identify the caller (access authorization and accounting) and characterize the job and the logging mode governing the job run.

The specifications regarding access authorization are checked against the entry in the JOIN file, additional specifications regarding job class and job attributes (job and run priority, system resources) are also checked against the entry in the job class definition. You can access these entries by means of the SHOW-USER-ATTRIBUTES or SHOW-JOB-CLASS command.

If the entries for PRIORITY and NTL (No Time Limit) in the JOIN file and the job class definition are not the same, the value which is most advantageous to you will be accepted.

The job runtime log is output to the SYSOUT (system) file. In addition, it may be specified that the log should also be written to SYSLST.

An (ENTER) file always begins with the LOGON command and ends with the LOGOFF command. The operands in the LOGON command are not evaluated, unless the (ENTER) file was started at the console. A command call error in an ENTER file usually triggers the SPIN-OFF mechanism (branch to the next ABEND, ABORT, STEP or LOGOFF command).

A copy of the file to be initiated is created under the name S.IN

- if the ENTER file is on private disk,
- if the ENTER file is cataloged under another user ID,
- if the ENTER file is a temporary file, or
- if the ENTER file is a library member.

After the ENTER file has been copied, the library member can be modified again. For this reason, the file at the time of ENTER may be different from the current one.

The S.IN file is automatically deleted at job end (LOGOFF) unless checkpoints were set during job execution (WRCPT macro, see "Executive Macros" manual [5]), in which case the S.IN.tsn file is required to ensure trouble-free restart (RESTART command)

Although S.IN... files are password-protected (EXPASS), it is possible to erase them with an /ERASE command without previously specifying the password. This makes it possible to remove from the system S.IN... files which are no longer needed or which were not deleted by the system itself.

ENTER files can be password-protected (CATALOG command) against reading (RDPASS), writing (WRPASS) and execution (EXPASS). The EXPASS password or a higher-ranking password must be specified in a PASSWORD command before specifying an ENTER command. If the file is to be deleted after execution (ENTER..., ERASE=YES), the WRPASS password must also be specified.

Passwords are checked during processing of the ENTER command. If a user subsequently changes the passwords, the access right still applies and the file is executed.

ENTER files may be SAM or ISAM files with variable record length (RECFORM=V). 72 characters are interpreted per data record. In the case of ISAM files, the key field may be located anywhere in the record because it is masked out.

The PRIORITY and MSG operands are still supported for reasons of compatibility only. Instead, you should use RUN-PRIO or RUN-PRIO in conjunction with START=IMMEDIATELY (for PRIORITY=(p,EXPRESS)) and LOG.

Fig. ENTER-1 Initiating an ENTER job

The following table contrasts the characteristics of ENTER and procedure files.


	ENTER file	Procedure file
How is the file or command sequence in the file called?	With the ENTER command	With the DO or CALL command
Is a new job created when the file is called?	Yes, a batch job	No: the DO or CALL command merely converts the procedure file to system file SYSCMD.
Which commands delimit the files?	LOGON command LOGOFF command	PROCEDURE command ENDP command
Can the file also be called within an ENTER file or a procedure file?	Yes	Yes
What happens when the file is called from a terminal?	The task sequence number (TSN) for the new batch job is displayed on the terminal. This job is then continued separately from the interactive job which has generated it.	The initiated procedure is part of the calling interactive job currently executing on the terminal. When all the commands in the procedure file have been processed, commands may be entered from the terminal again.
Can symbolic operands whose values are defined only when the file is called or executed be used in the file?	No	Yes

### Format and operand description


Name	Operation	Operands
[. jobname]	<pre>{ ENTER } { E }</pre>	<pre>pathname  [,userid1,accountno[,password]]  [,FPASS=password]  [,ERASE={ NO } YES ]  [,HOST={ 'hostid' } jvname1 ] [,CAT={ 'catid' } jvname2 ]  [,JOB-CLASS={ jobclass } *STD ]  [,MONJV=jvname] [,JVPASS=password]  [,JOB-PRIO={ jprio } STD ]  [,RERUN={ NO } YES ]  [,FLUSH={ NO } YES ]  [ , START= { STD SOON IMMEDIATELY WITHIN ( { HOURS=hour [, MINUTES=minute] [HOURS=hour, ]MINUTES=minute } ) } ]  AT ( [DATE=yy-mm-dd, ]TIME=hh:mm) EARLIEST ( [DATE=yy-mm-dd, ]TIME=hh:mm) LATEST ( [DATE=yy-mm-dd, ]TIME=hh:mm) [AT-STREAM-STARTUP ]</pre>


Name	Operation	Operands
	<pre> {ENTER} {E} (cont.)                     </pre>	<pre> {STD NO DAILY WEEKLY [, REPEAT= { PERIOD( {HOURS=hour[, MINUTES=minute]} ) {[HOURS=hour, ]MINUTES=minute} AT-STREAM-STARTUP } [, RUN-PRIO= {rprio} {STD} ] [, TIME= {t} {NTL} {STD} ] [, PRINT= {number} {NO} {STD} ] [, PUNCH= {number} {NO} {STD} ] [, LOG= (LISTING= {NO} {YES} ) ] [, JOB-PAR= {*NO} {string} ] [, PRIORITY= {p} {([p], EXPRESS)} ] [, MSG= {F} {C} ] [L] [H]                     </pre>

jobname	<p>Defines a 1- to 8-character job name (for valid characters see page 7) which is printed in uppercase letters (including user ID and account number) in job listings and on any spool jobs initiated by the job.</p> <p>"jvname" is also the label used for branching to the command.</p>
pathname	<p>stands for: <code>[:catid:][\$userid.]</code> <math>\left\{ \begin{array}{l} \text{filename} \\ \text{library(member)} \end{array} \right\}</math></p>
catid	<p>Catalog ID of the pubset on which the file is stored. Default value: the catalog ID assigned to the user ID (JOIN entry).</p>
userid	<p>User ID to which the file is assigned.</p> <p>Default value: the user ID from the LOGON command.</p>
filename	<p>Name of the cataloged file with the batch job.</p> <p>If the file is not cataloged under your user ID, it must be shareable and the user ID under which it is cataloged must be specified.</p> <p>The (ENTER) file must also be shareable if the task calling the ENTER command is running under a user ID other than the one specified in the ENTER command. In this case, "password" must be specified if "userid1" is password-protected. An execute password (EXPASS) on the file must be specified with the FPASS operand or beforehand with the PASSWORD command.</p> <p>"filename" may also be the name of a temporary file (see the manual "DMS Introductory Guide and Command Interface" [8].</p> <p>File generations or file generation groups are not allowed.</p> <p>Specification of a file group ("file(group)") is only allowed for tape files.</p>
library	<p>Name of an LMS library on disk (see the "LMS" manual [14]).</p>
(member)	<p>Name of the library member with the batch job. "member" may be up to 38 characters in length:</p> <ul style="list-style-type: none"> <li>– letters                                   A, ..., Z</li> <li>– special characters                       \$, #, -, ., @</li> <li>– digits                                     0, ..., 9</li> </ul> <p>The first character must be a letter. The last character must not be a hyphen.</p> <p>The expression "library(member)" may be up to 42 characters in length minus the length of the catalog ID (including parentheses). "library" or "member" may be up to 38 characters in length.</p>

userid1	User ID for the ENTER job to be initiated.
accountno	Account number for the ENTER job.
password	LOGON password for "userid1". The password cannot be logged to SYSOUT, i.e. it does not appear in the printer listing for the ENTER job.  "userid" and "accountno" may only be specified or omitted together in the ENTER command.  A status check (STATUS command) is only possible under the user ID of the calling job.  If "userid1", "accountno" and "password" are omitted, they will be copied from the LOGON command of the initiating job.
CAT	Specifies a catalog ID. The job is directed to the computer to which the specified catalog is assigned. If both HOST and CAT are specified, the HOST operand is used to determine the host computer.
= 'catid'	Catalog ID. "catid" must be a catalog which is known to and accessible in the multiprocessor network, otherwise the ENTER command will be rejected.
= jvname2	Job variable containing "catid". "jvname2" must contain the "catid" of a catalog which is known and accessible in the multiprocessor network, otherwise the ENTER command will be rejected. If "jvname1" or "jvname2" is not accessible, the ENTER command is rejected. The syntax for "jvname1"/"jvname2" must be in accordance with the conventions for a GETJV operation.
	 HOST and CAT are only available with the software products MSCF and JV.



ERASE	Specifies that the ENTER file should not be erased at the end of the job (default value).
= <u>NO</u>	
=YES	Specifies that the ENTER file should be erased at the end of the ENTER job.
	 ERASE=YES will be ignored if <ol style="list-style-type: none"> <li>a) the file is a library member,</li> <li>b) the file is not cataloged under the caller's user ID,</li> <li>c) the job is terminated abnormally,</li> <li>d) the job is aborted by the ABEND, CANCEL or SHUTDOWN command.</li> </ol> <p>c) and d) do not apply if:</p> <ul style="list-style-type: none"> <li>– the file is on private disk,</li> <li>– the file is a temporary file,</li> <li>– the file is cataloged under the caller's user ID and the ENTER job is to be executed under another user ID.</li> </ul> <p>In such cases the file is erased after creation of the S.IN file.</p>

FLUSH	
= <u>YES</u>	The job will be removed from the job queue if it has not been processed by the end of the BS2000 session (SHUTDOWN).
	 The RERUN and FLUSH operands are not analyzed for a repeat job. The job is initiated at the next repetition point.
= <u>NO</u>	The job will remain in the job queue. The next session must be initiated with a warm start or selective start (default value).

### Job control with RERUN/FLUSH

- If FLUSH=YES and RERUN=YES were specified and the job was interrupted in a previous session, then FLUSH=NO will be used in the next session. This ensures that the job remains in the job queue, even if it is not initiated in this session.
- A monitoring job variable is set to "\$S" when the job is repeated.
- RERUN and FLUSH are not analyzed for repeat jobs.

FPASS=password

Indicates the execute or write password for the ENTER file:  
 A write password is only needed if ERASE=YES is specified. The password (see CATALOG command: "passwords" for syntax) for the ENTER file must be specified in the ENTER command if job distribution is requested (see "MSCF Multiprocessor System" manual [15]). In the absence of job distribution, the password can also be specified by means of a separate PASSWORD command.

HOST

= "hostid"

Indicates the host computer on which the job is to be executed. "hostid" must refer to an active computer within the multiprocessor network, otherwise the ENTER command will be rejected. If both HOST and CAT are specified, the HOST operand is used to determine the host computer.

= jvname1

Job variable containing "hostid". "jvname1" must contain the host ID of an active computer within the multiprocessor network, otherwise the ENTER command will be rejected.

The syntax for "jvname1"/"jvname2" must be in accordance with the conventions for a GETJV operation.



HOST and CAT are only available with the "MSCF" and "JV" software products.

JOB-CLASS

Denotes a job class into which the job is to be incorporated. Authorization for the various job classes can be displayed by means of the SHOW-USER-ATTRIBUTES or SHOW-JOB-CLASS commands.

= jobclass

Name of job class.

= \*STD

(Default) job class preset by the user or the operating system; default value

JOB-PAR

Enables specification of additional attributes for the selected job class, provided the system administrator has defined and announced such attributes.

= \*NO

No additional attributes; default value.

= "string"


Any sequence of characters; it is provided by the system administrator for identifying additional job class attributes.

JOB-PRIO	Determines the priority (in relation to other jobs) for initiation of a batch job. This specification does not affect subsequent execution of the job.
=jprio	Job priority. $\text{MAXIMUM} \leq \text{jprio} \leq 9$ . The lower the value, the higher (greater) the priority (urgency). The MAXIMUM value is defined in the job class definition and can be displayed on the screen by means of the SHOW-JOB-CLASS command. If MAXIMUM=NO or no value is specified, the default priority is assumed.
= <u>STD</u>	The default value for the job class is used; default
JVPASS	Denotes the password for access to the monitoring job variable, in accordance with the password hierarchy. The password must be specified in the ENTER command if job distribution (see "MSCF Multiprocessor System" manual [15]) is requested. In the absence of job distribution, the password can also be specified via a separate PASSWORD command.  JVPASS is ignored if MONJV was not specified.
=password	Password for the "jvname" job variable.
LOG=(...)	Specifies whether or not the job runtime log will also be output to SYSLST.
LISTING=YES	The log will also be output to SYSLST.
LISTING= <u>NO</u>	The log will not be output to SYSLST; default value.
MONJV	Indicates a job variable that is to monitor the job. The user can address his job via this job variable. During job execution, the operating system assigns the following values to the job variable: <ul style="list-style-type: none"> <li>– \$S (job in job queue),</li> <li>– \$R (job being processed),</li> <li>– \$T (job terminated normally) or</li> <li>– \$A (job aborted).</li> </ul> <p>When ENTER is executed, the status display for "jvname" is set to "\$S", the "tsn" display is set to the task number (TSN) for the job, and the processor display is set to the catalog ID of the processor on which the job is to be executed.</p>

If "jvname" is not accessible at command execution time, an error message is output to SYSOUT and command execution is continued without monitoring.

Both the user ID issuing the monitoring job variable and the user ID for which the job is being processed must be able to access "jvname".

Access to the monitoring job variable is subject to the same conventions as access to the ENTER file.

 The MONJV operand is only available with the JV software product (see also "Job Variables" manual [11]).

=jvname      Name of job variable.


MSG      Defines the type of logging for the remainder of job execution. Operand MSG is supported for reasons of compatibility only. MSG is completely ignored if LOG has been specified.


=F      System messages are output in full-length form to the system file SYSOUT (F stands for "full message"); default value.

=C      The coded short form of system messages is output to SYSOUT (C stands for "code").

=L      Console messages and operator responses are logged to SYSOUT for this job.  
Operator commands used for controlling job execution (e.g. changes in priority) as well as general (system) warnings and error messages for the operator are not output (L stands for "log").  
If the user enters MSG=LH, the messages logged to SYSLST will additionally be given the time of day at which they were issued.


=H      All messages written to SYSOUT are also written to SYSLST (H stands for "hold message"). Exception: system messages which require a response from the user, as well as the message ABNORMAL PROGRAM TERMINATION.

 In line mode (MODE=LINE), logging takes place line by line, i.e. control character NL is analyzed. In format mode (MODE=FORM), logging is continuous, i.e. the log is not represented true to format and control character NL is not analyzed.

PRINT	<p>Indicates the maximum number of records which a task can output (in summary form) to the system files SYSLST, SYSLST01, SYSLST02 through SYSLST99. Records written simultaneously to the system files SYSOUT and SYSLST (LOG=LISTING or MSG=FUH operands) are not included.</p> <p><i>Notes on exceeding the specified number:</i></p> <ul style="list-style-type: none"> <li>– in batch mode the task terminates abnormally;</li> <li>– in interactive mode, the user may specify whether the task should continue or be terminated. If continued, it will again be output until "number" is reached.</li> </ul> <p>=number      Number of records, where <math>0 \leq \text{number} \leq 999999</math>.</p> <p>=NO            Number of records is unlimited.</p> <p>=<u>STD</u>        Default value of the selected job class; preset value.</p>
PRIORITY	<p>Determines the priority (relative to other tasks) for executing the job. PRIORITY is ignored if the RUN-PRIO operand has been executed.</p> <p> PRIORITY is only supported for reasons of compatibility. You should use RUN-PRIO instead.</p> <p>=p             Run priority. <math>\text{MAXIMUM} \leq p \leq 255</math>.</p> <p>=(p,EXPRESS) EXPRESS causes the ENTER job to be started immediately. It has no further effect on job execution. EXPRESS is ignored if the START operand has been specified.</p>
PUNCH	<p>Indicates the maximum number of records output by the task to the system file SYSOPT.</p> <p>=number      Number of records, where <math>0 \leq \text{number} \leq 999999</math>.</p> <p>=NO            Number of records is unlimited.</p> <p>=<u>STD</u>        Default value of the selected job class; preset value. Refer to the PRINT operand for information on exceeding the specified number of records.</p>
REPEAT	<p>Indicates an interval of time after which the job is to be periodically started. The repetition is considered as a job sequence. J(0) indicates the first job run, J(1) the first repetition, ..., J(n) the nth repetition of the job. With the start of job J(i), the repetition J(i+1) is also created, where <math>i \geq 0</math>.</p>


- =STD Default value of the selected job class; preset value.
- =NO The job will not be repeated.
- =DAILY Daily repetition at the time-of-day specified with START.
- =WEEKLY Weekly repetition at the time-of-day specified with START.
- =PERIOD(...) Repetition following the specified interval of time (in hours and minutes).  
 $0 \leq \text{hour} \leq 23; 0 \leq \text{minute} \leq 59.$

=AT-STREAM-STARTUP Repetition following job scheduling startup.

 Specification of the repeat values (NO, DAILY, WEEKLY, PERIOD and AT-STREAM-STARTUP) is permitted only if they are also permitted in the job class definition (see also SHOW-JOB-CLASS command).

RERUN

=YES The job will be reinitiated in the next BS2000 session if execution was interrupted by an unrecoverable system error or the end of the session.


 The operands RERUN and FLUSH are not analyzed in the case of a repeat job. The job is started at the time of the next repeat.


=NO The job will not be reinitiated.

RUN-PRIO Indicates the urgency (relative to other tasks) for processing the job.

=rprio Run priority, where  $\text{MAXIMUM} \leq \text{rprio} \leq 255.$   
 The lower the value, the higher (greater) the priority (urgency). The MAXIMUM value is defined both in the job class definition and in the JOIN file, and can be interrogated by means of the SHOW-JOB-CLASS or SHOW-USER-ATTRIBUTES command. If the values do not match, the limit value most favorable to the job is used.

=STD Default value of the selected job class; preset value.

 The default value is also used when illegal values are specified for "rprio".

- START** Specifies a time (period) for starting a job.
- Values entered for start time:  
 DATE = yy-mm-dd : (yy=year, mm=month, dd=day).  
 TIME = hh:mm : time-of-day (hh=hour, mm=minute).
- Hyphens or the colon must be included in the entry for DATE or TIME.
- Example: 31st May 1990 at 15.08 hours  
 AT (DATE=90-05-31, TIME= 15:08).
- For TIME:  $00 \leq hh \leq 23$ ;  $00 \leq mm \leq 59$ .
- The ENTER command is rejected if the month or time-of-day specified is in the past. However, specification of a past year "yy" is interpreted as the year 20yy.
-  The default values SOON, IMMEDIATELY, WITHIN, AT, EARLIEST, LATEST and AT-STREAM-STARTUP may only be specified if they are also permitted in the job class definition; (see also SHOW-JOB-CLASS command).
- =STD** The default value for the selected job class is used; preset value.
- =SOON** The job is to be started as soon as possible, taking its priority into account.
- =IMMEDIATELY** The job is to be started immediately.
- =WITHIN(...)** The job is to be started within the specified time (in hours and minutes).  
 $0 \leq \text{hour} \leq 23$ ;  $0 \leq \text{minute} \leq 59$ .
- =AT(...)** The job is to be started at exactly the time (date, time-of-day) specified.
- =EARLIEST(...)** The job is to be started on or after the specified time (date, time-of-day).
- =LATEST(...)** The job is to be started on or before the specified time (date, time-of-day).
- =AT-STREAM-STARTUP** The job is to start following job scheduler startup.

- TIME** Denotes the maximum CPU time (in seconds) permitted for the job. The maximum CPU time which can be specified here is determined by the job class selected.
- =t CPU time in seconds, where  $0 \leq t \leq$  maximum CPU time.
  - =NTL NTL: No Time Limit. The task will run with unlimited CPU time.
  - =STD Default value of selected job class; preset value.

**Combinations of the START and REPEAT operands:**

START	REPEAT		
	AT-STREAM-STARTUP	DAILY or WEEKLY	PERIOD
IMMEDIATELY or SOON	a)	c)	c)
AT or EARLIEST	a)	d)	f)
LATEST or WITHIN	a)	c)	g)
AT-STREAM-STARTUP	b)	e)	h)

- a) The first and all subsequent job starts occur as specified.
- b) The first job start occurs with START=AT-STREAM-STARTUP. All subsequent starts occur after job scheduler startup with START=SOON.
- c) Time base for the repeat cycle is the time the job was accepted.
- d) The specified time (START=..., TIME=...) is the time base for the repeat cycle.
- e) The first job start occurs after the job scheduler startup. This start time is the time base for the repeat cycle. The subsequent starts are effected with START=SOON.
- f) The specified time (START=..., TIME=...) is the time base for the repeat cycle. The second and all subsequent starts occur with START=SOON.
- g) The time base for the repeat cycle is the time the job was accepted. All subsequent starts occur with START=SOON.



- h) Time base for the repeat cycle is the first start time. The first job start occurs after the job scheduler startup. The subsequent starts occur with START=SOON.
- The nth repetition ( $n \geq 1$ ) of a job only starts when the (n-1)th has finished.
  - Aborting the currently running job J(n) has no effect on the start of J(n+1), where  $n \geq 0$ .
  - Aborting the entire job: Both the currently running job J(n) and the subsequent job J(n+1) are terminated where  $n \geq 0$  (CANCEL command or make job J(n) the last one in the repeat sequence with command MODIFY-JOB tsn, REPEAT=NO).

## Examples

### Example 1

File X.Entering.1 contains the following:

```
/LOGON
/ESTATUS
/FSTAT ,ALL
...
/LOGOFF
```

The following listing is produced in interactive mode:

```
.....
(IN) FSTAT X.Entering.1
(OUT) 0000003 :N:$PA123456.X.Entering.1
      :N: PUBLIC: 1 FILE. RES= 3, FREE= 2, REL= 0 PAGES

(IN) ENTER X.Entering.1,TIME=155 _____ (01)
(OUT) % JMS0066 JOB 'SALEM' ACCEPTED ON 90-12-18 AT 09:32, TSN = 0YAY.

(IN) STA T=0YAY,DISP=L
(OUT) NAME TSN TYPE PRI CPU-USED CPU-MAX ACCOUNT#
      SALEM 0YAY 1 WT 9 230 0.0 155 M1234MON
```

- (01) The ENTER command for file X.Entering.1 initiates a new batch job (TSN=0YAY). The STATUS command is subsequently used to provide information on the job status.

Example 2: Specifying the TIME operand for ENTER jobs

```
/LOGON PA123456,M1234MON,TIME=50
```

```
(IN) ENTER X. ENTER.1 _____ (01)
(OUT) % JMS0066 JOB 'SALEM' ACCEPTED ON 90-12-18 AT 09:46, TSN = 0YDB.
```

```
(IN) STA L
(OUT) NAME TSN TYPE PRI CPU-USED CPU-MAX ACCOUNT#
      SALEM 0YDB 1 WT 9 230 0.0 2000 M1234MON
      SALEM 0Y CZ 3 DIALOG 0 210 0.4135 50 M1234MON
(OUT) % SPS0171 NO LOCAL SPOOLOUT JOB PRESENT
      % SRO0376 NO RSO JOB OF TYPE=7 PRESENT
```

```
(IN) ENTER X. ENTER.1, TIME=60 _____ (02)
(OUT) % JMS0066 JOB 'SALEM' ACCEPTED ON 90-12-18 AT 09:47, TSN = 0YDF.
```

```
(IN) STA L
(OUT) NAME TSN TYPE PRI CPU-USED CPU-MAX ACCOUNT#
      SALEM 0YDB 1 WT 9 230 0.0 2000 M1234MON
      SALEM 0YDF 1 WT 9 230 0.0 60 M1234MON
      SALEM 0Y CZ 3 DIALOG 0 210 0.7447 50 M1234MON
(OUT) % SPS0171 NO LOCAL SPOOLOUT JOB PRESENT
      % SRO0376 NO RSO JOB OF TYPE=7 PRESENT
```

```
/LOGOFF BUT
```

```
...
...
...
```

```
/LOGON PA123456,M1234MON
```

```
(IN) ENTER X. ENTER.1 _____ (03)
(OUT) % JMS0066 JOB 'SALEM' ACCEPTED ON 90-12-18 AT 10:00, TSN = 0Z19.
```

```
(IN) STA L
(OUT) NAME TSN TYPE PRI CPU-USED CPU-MAX ACCOUNT#
      SALEM 0YDB 1 WT 9 230 0.0 2000 M1234MON
      SALEM 0Z19 1 WT 9 230 0.0 2000 M1234MON
      SALEM 0Z11 3 DIALOG 0 210 0.4053 32766 M1234MON
(OUT) % SPS0171 NO LOCAL SPOOLOUT JOB PRESENT
      % SRO0376 NO RSO JOB OF TYPE=7 PRESENT
```

(01) If the TIME operand is omitted from the ENTER command, the ENTER job will be given the default value from the assigned job class.

- (02) The TIME specification in the ENTER command was assumed for the ENTER job with TSN 0YDF.
- (03) A new interactive job begins. The ENTER job with TSN 0Z19 receives the default value from the assigned job class as TIME specification, as does the new interactive job.  
The ENTER job with TSN 0YDF is no longer in the system.

For further examples, see the LOGON and SKIPUS commands.

## EOF Indicate end-of-file for SYSDTA

Application group: Program control (page 30)

### Command description

The EOF command generates an EOF condition for the system input file SYSDTA. It is used in conjunction with an RDATA macro (see the "Executive Macros" manual [5]) and a corresponding language statement in the user program. This macro reads a data record from the system file SYSDTA. If the record read is an EOF command, control is passed to the routine for end-of-file processing in the user program; otherwise, the program continues with the command following the RDATA macro. The address of this routine must have been specified as an operand in the RDATA macro.

The EOF condition is removed if

- the /SYSFILE SYSDTA = ... command is entered, or
- if the program terminates abnormally.

In interactive mode, the EOF command can be used in conjunction with the ESCAPE function to terminate data readin at the terminal (WRTRD macro, see the "Executive Macros" manual [5]).

### Format

Operation	Operands
EOF	

**Examples:**

*Example 1:*

The following command/data sequence is entered for a batch job:

```

/LOGON ...
/EXEC X.PROCESS _____ (01)
RECORD1      }
RECORD2      } Data for program X.PROCESS
.....      }

/EOF _____ (02)
/LOGOFF

```

- (01) Program X.PROCESS reads in data from the SYSDTA system file with the aid of the RDATA macro (see chapter 1). Since SYSDTA and SYSCMD are merged in the batch job, the data is expected immediately after the EXEC command.
- (02) As soon as the EOF command is encountered, program X.PROCESS branches to its end-of-file processing routine.

*Example 2:*

If the data for program X.PROCESS (see Example 1) is to be entered on the terminal, the listing is as follows:

```

(IN)   /EXEC X.PROCESS
(OUT)  % BLS0500 PROGRAM PROCESS VERSION OF 85-08-16 LOADED
        .RECORD1
        .RECORD2
        .....
_____  

(IN)   /EOF _____ (01)
        / _____ (02)

```

- (01) System mode is entered by means of the ESCAPE function (e.g. K2 key at the terminal), causing the system to display a slash.
- (02) The EOF command indicates end-of-data for the input file. The program branches to the end-of-file routine specified in the RDATA call.

## ERAJV Delete job variable

Application group: Job variable functions (page 36)

The ERAJV command is only available with the JV software product (see also "Job Variables" manual [11]).

### Command description

The ERAJV command erases one or more job variables in the file catalog or overwrites the job variable values with X'00...00'. The job variable names may be specified in partially qualified form or contain wildcards.



Job-monitoring job variables are protected by the system and may not be deleted.

### Format and operand description

Operation	Operands
ERAJV	<pre> {pathname} {*jvlink} [, CHECK={   STD   NO   MULTIPLE   PVS   SINGLE }] [#]  [, DATA={   NO   YES }] [, IGNORE={   NONE   protect1   (protect1, ...4) }]  [, PASSWORD={   NONE   password1   (password1, ...3) }]  [, LIST={   YES   NO }] </pre>

pathname stands for: [:catid:][userid.][jvname]

"pathname" is the job variable to be erased. You can erase your own job variables only. Wildcards may be specified for "catid", "userid" and "jvname" (cf. the FSTATUS command).

catid Catalog ID of the pubset containing the job variable. Default value: the catalog ID (JOIN entry) assigned to the user ID.

userid	User ID. Only the user ID of the current job may be specified. Default value: user ID from the LOGON command. Specification of the user ID is mandatory if a blank is specified for "jvname".
jvname	Name of the job variable to be erased. It can be specified in fully or partially qualified form or using wildcards. The names of temporary job variables must start with a prefix.
*jvlink	Specifies the job variable to be erased via its JV link name. Wildcards are not allowed for "*jvlink".
#	Erases all temporary job variables of the job. "#" stands for the special character defined using the system parameter TEMPFILE for identifying temporary files and job variables.
CHECK	Erasing the job variable is made subject to the user's response to a query. This operand is ignored in batch mode.
=STD	Is the default value for interactive applications or for procedures and in batch mode. The following applies: MULTIPLE for interactive mode, NO for procedures and batch mode.
=NO	The catalog entry or the job variable value is erased without an acknowledgment. This is the default value for batch mode and procedures.
=MULTIPLE	Results in a global query for each user ID (message: JVS0465), but only if more than one job variable is to be erased. MULTIPLE is the default value in interactive mode.
=PVS	Results in a global query for each pubset (message: JVS0468), but only if more than one job variable is to be erased.
=SINGLE	A query is made for each job variable that is to be erased (message: JVS0469).  With MULTIPLE, PVS and SINGLE, the following query is appended to each message: REPLY (Y=YES; N=NO; T=TERMINATE; [,CHECK=NEW MODE])

Response	Effect
Y	The erase request indicated is executed.
N	The erase request indicated is not executed (message JVS046A).
T	The erase procedure is aborted at the erase request indicated.
any other response	Same effect as N.
CHECK=	The current check mode is not modified.
CHECK= <u>new</u> mode	The new check mode (SINGLE / MULTIPLE / PVS / STD / NO takes effect immediately; response Y/N is not analyzed and can therefore be omitted.
CHECK= <u>old</u> mode	Response Y/N is analyzed; this is also true for <u>old</u> mode = MULTIPLE and <u>new</u> mode = STD.

- DATA** Determines whether the catalog entry is to be erased or the job variable value is merely to be overwritten with X'00...00'.
- =NO The catalog entry is to be erased; default value.
  - =YES The job variable value is to be overwritten with X'00...00'; the catalog entry is not erased.
- IGNORE** Specifies whether protection attributes entered in the catalog for the job variable are to be ignored when the catalog entry is erased.
- =NONE Protection attributes are not ignored; default value.
  - =protect1,..4 Specifies the protection attribute to be ignored. The following values can be specified individually or in a list of up to four values:
    - ACCESS: The protection attribute ACCESS=READ and the access rights assigned with BASIC-ACL are ignored.
    - EXDATE: An expiration date later than the date of the current day is ignored.



---

LIST	Specifies whether the job variables erased are to be logged.
= <u>NO</u>	Logging is suppressed; default value.
=YES	All job variables erased without error are logged.
PASSWORD	Enables the user to erase password-protected job variables. Passwords specified here are not entered in the task password table.
= <u>NONE</u>	Prevents password-protected job variables from being erased; default value.
password1,...	Indicates the password(s) to be ignored when job variables are erased. Up to three passwords can be specified in a list.

Refer to "Job Variables" manual [11] for examples.

## ERASE Erase file

Application group: File processing (page 26 ff.)

### Command description

The ERASE command enables you to erase

- files,
- file generation groups,
- file generations,
- temporary files and
- system files for output

The file to be deleted must be cataloged under your user ID.

The following types of deletion are possible:

- a) Logical deletion of a file ("pathname" operand):  
The catalog entry is deleted and the allocated storage space released. Data bytes are not overwritten with X'00'. Example: /ERASE file
- b) Physical deletion of a file (DESTROY operand):  
The catalog entry is deleted and the allocated storage space released. The data bytes are overwritten with X'00'. Example: /ERASE file,DESTROY  
A file is also physically deleted although the DESTROY operand is omitted if DESTROY=YES has been specified in its catalog entry (see the CATALOG command).
- c) Logical deletion of the data of a file (DATA/SPACE operand):  
The user is no longer able to access the contents of a file. The data bytes are not overwritten with X'00'. The catalog entry is retained. When the DATA operand is specified, the storage space remains allocated; when SPACE is specified, the storage space is released. Example: /ERASE file,DATA
- d) System files for output (SYSLST/SYSLSTn/SYSOUT/SYSOPT operands):  
Only the contents of the file are logically deleted, while the catalog entry S.LST./OUT./OPT. .... is retained until the end of the task. If the file is assigned to a cataloged (user) file, the contents of the latter are logically deleted, the assignment is retained. Example: /ERASE \*SYSLST

The operands for the ERASE command can be divided into four groups corresponding to the four functional levels.

- Selection operands

The selection operands serve to specify the files or catalog entries to be processed, using attributes stored in the catalog entries as selection criteria. For this reason, some operands of the FSTATUS command can also be specified in the ERASE command.

Attribute	Operand	Operand value	Function
File name	pathname		Specifies the catalog entries to be processed (fully-qualified, partially qualified, with wildcards)
	prefix		Prefix of all temporary user files
	*		Refers to the EAM object module file
	*SYSfile		Refers to a system file (SYSLST, SYSOUT, ...); wildcards are permitted
	*DUMMY		Refers to a dummy file
File type	TYPE	FILE	No FGGs or file generations are processed
		FGG	Only file generations or FGGs are processed
	POS	AFTER/ BEFORE	Specified with TYPE=FGG; specifies the file generations to be processed
	FCBTYPE	NONE	Only catalog entries of unopened files, i.e. of files existing purely as catalog entries, are processed
		ISAM/SAM/ BTAM/PAM	Files to be processed are selected via their access method
	BLKCTRL		Files to be processed are selected via their file format
Volume	SUPPORT	PUBLIC	Only files on public disk are processed
		PRDISC	Only files on private disk are processed
		TAPE	Only tape files are processed
	VOLUME	vsn	Only files on the volume with volume serial number "vsn" are processed

Attribute	Operand	Operand value	Function
Migration	MIGRATE	ALLOWED	Only files which are allowed to migrate are processed
		INHIBIT	Only files which are not allowed to migrate are processed
	STORAGE-LEVEL	S0/S1/S2	Only files on the specified storage level(s) are processed
Data security and protection	BACKUP	A/B/C/D/E	Files to be processed are selected via the backup level
	ACCESS	READ	Only files with write protection (i.e. ACCESS=READ) are processed
		WRITE	Only files without write protection are processed
	SHARE	NO	Shareable files are not processed
		YES	Only shareable files are processed
	PASS	SPECIAL	Only files which can be accessed under the maintenance user ID are processed.
		NONE	Only files without password protection are processed
EXPASS RDPASS WRPASS		Files to be processed are selected via the type of password protection defined for them	
Storage space	EXTENTS		Files to be processed are selected via the number of extents they occupy
	FRESIZE		Files to be processed are selected via the size of reserved but unused storage space
	SIZE		Files to be processed are selected via the size of reserved storage space
Dates	CRDATE		Files to be processed are selected via their creation date
	EXDATE		Files to be processed are selected via their expiration date (which implies their retention period)
	LADATE		Files to be processed are selected via their date of last access

- File protection operands

File protection operands obviate entry of the CATALOG or PASSWORD commands in order to erase files for which file protection attributes such as passwords, retention period, etc. are defined.

Operand	Operand value	Function
IGNORE	NONE	Protection attributes are evaluated
	ACCESS	The protection attribute ACCESS=READ and file protection by BASIC-ACL are ignored
	EXDATE	Retention periods are ignored
PASSWORD	NONE	Password protection is observed
	password	Protection by the specified password is ignored

- Action operands (operands for command execution)

Action operands control the internal execution of the ERASE command. They enable you to specify not only the scope but also the conditions of deletion.

Operand	Function
SPACE-CATALOG	The catalog entry is deleted and storage space released
SPACE	Only storage space is released, the catalog entry is retained
DATA	Logical erase: the last-page pointer is reset, the catalog entry and the allocated storage space are retained
CATALOG	Files on private volumes are exported
DELETE-OR-EXPORT	Files on private volumes are exported, files on public volumes are deleted
DESTROY	The catalog entry is deleted, the storage space is released and cleared
MOUNT	Specifies, for files on private disk, whether all affected disks must be online

- Control operands

Control operands enable you to define the user interface yourself within certain limits. In interactive mode, for example, you can use prompting, request a SYSOUT listing or have DMS errors ignored (as regards SPIN-OFF).

Operand	Operand value	Function
CHECK	NO	All files are erased by ERASE; access is no longer possible (default value for procedures)
	MULTIPLE	Dialog when the catalog or user ID is changed if "pathname" was not fully-qualified (default value for interactive mode)
	ERROR	Switchover from CHECK=NO to dialog if an error occurs
	PVS	Dialog when the catalog ID is changed if "pathname" was not fully-qualified
	SINGLE	You decide interactively for each file whether it is to be processed by the current ERASE operation
LIST	NO/YES	Erasure is (not) to be logged on SYSOUT
NOSTEP	NONE	Errors during command execution trigger a SPIN-OFF
	errcode	You can specify, via the DMS error code, which errors are not to trigger a SPIN-OFF

### Format and operand description

Operation	Operands
ERASE	<pre> {pathname prefix [* *SYSfile *DUMMY ]}  [,TYPE={ANY FILE FGG}] [,POS={AFTER BEFORE}]  [,FCBTYPE={ANY ISAM BTAM (SAM, ...) PAM NONE}] [,BLKCTRL={ANY PAMKEY DATA (NO, ...) NONE}]  [,SUPPORT={ANY PUBLIC (PRDISC, ...) TAPE}] [,VOLUME={ANY vsn}] [,BACKUP={ANY A B C, ... D E}]  [,ACCESS={ANY READ WRITE}] [,SHARE={ANY NO YES SPECIAL}] [,PASS={ANY NONE EXPASS (RDPASS, ...) WRPASS}]  [,EXTENTS={ANY number (number[,]) (number1,number2)}] [,FREESIZE={ANY SIZE number (number[,]) (,number) (number1,number2)}] [,SIZE={ANY F[REE]SIZE number (number[,]) (,number) (number1,number2)}]  [,CRDATE={ANY NONE date (date[,]) (,date) (date1,date2)}] [,EXDATE={ANY NONE date (date[,]) (,date) (date1,date2)}] [,LADATE={ANY NONE date (date[,]) (,date) (date1,date2)}] </pre>

Operation	Operands
ERASE (cont.)	$[ , \text{MIGRATE} = \left\{ \begin{array}{l} \text{ANY} \\ \left\{ \begin{array}{l} \text{ALLOWED} \\ \text{INHIBIT} \end{array} \right\}, \dots \end{array} \right\} ] [ , \text{STORAGE-LEVEL} = \left\{ \begin{array}{l} \text{ANY} \\ \left\{ \begin{array}{l} \text{S0} \\ \text{S1} \\ \text{S2} \end{array} \right\}, \dots \end{array} \right\} ]$ $[ , \text{IGNORE} = \left\{ \begin{array}{l} \text{NONE} \\ \left\{ \begin{array}{l} \text{ACCESS} \\ \text{EXDATE} \end{array} \right\}, \dots \end{array} \right\} ] [ , \text{PASSWORD} = \left\{ \begin{array}{l} \text{NONE} \\ \text{(password, ...)} \end{array} \right\} ]$ $[ , \left\{ \begin{array}{l} \text{SPACE-CATALOG} \\ \text{SPACE} \\ \text{DATA} \\ \text{CATALOG} \\ \text{DELETE-OR-EXPORT} \\ \text{DESTROY} \end{array} \right\} ] [ , \text{MOUNT} = \left\{ \begin{array}{l} \text{FIRST-DISK} \\ \text{ALL-DISKS} \end{array} \right\} ]$ $[ , \text{CHECK} = \left\{ \begin{array}{l} \text{STD} \\ \text{NO} \\ \text{MULTIPLE} \\ \text{ERROR} \\ \text{PVS} \\ \text{SINGLE} \end{array} \right\} ] [ , \text{LIST} = \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} ] [ , \text{NOSTEP} = \left\{ \begin{array}{l} \text{NONE} \\ \text{(errcode, ...)} \end{array} \right\} ]$

**Positional operands**

- pathname            stands for [:catid:][\${userid.}][filename]

"pathname" designates the files which are to be erased. You can only erase your own files. Wildcards may be specified in "catid" and "filename" (cf. the FSTATUS command). If the wildcard "\*" is used, it must be entered twice ("\*\*") if it is to include the beginning of a file name.
- catid                Catalog ID of the pubset containing the file. If wildcards are used in the catalog ID they are evaluated for the local system only. Catalogs from a foreign system can be addressed only via an explicit "catid". Default value: the catalog ID assigned to the user ID (JOIN entry).
- userid              User ID assigned to the file. Only the user ID of the current job may be specified. Default value: the user ID from the LOGON command.



filename	Designates the files, file generations, FGGs or temporary files to be erased. You may specify a fully or partially qualified file name or use wildcards. The prefix must be included in the names of temporary files; otherwise, temporary files will be ignored.
prefix	All temporary files of the job can be erased by specifying the prefix defined for temporary files. If the erase operation is logged or if it is executed via the system dialog, the internal names of the temporary files being processed are displayed on the screen.
*	The ERASE command refers to the EAM object module file (* file) which is created and used by the language processors. All operands except the control operands (CHECK, LIST, NOSTEP) are checked for formal errors only; otherwise they are ignored when the command is executed. Errors which occur when the * file is erased are ignored.
*SYSfile	<p>Designates the system files SYSLST, SYSLSTn, SYSOUT and SYSOPT (file = LST, LSTn, OUT, OPT; n = 01, 02, ..., 99). The contents of the specified system file or the (user) file to which it is assigned are logically deleted. The file is again written to, starting at the beginning of the file; any previous assignment is retained.</p> <p>Wildcards may be specified for "file", which means that one ERASE command might be used for several system files. All operands except the control operands (CHECK, LIST, NOSTEP) are checked for format errors but otherwise ignored during execution.</p> <p>Specifying *SYSOUT is permitted only in batch mode; in interactive mode, it causes a message to be issued, and in procedures, the spin-off mechanism is triggered. If ERASE *SYSLST (or ERASE *SYSOUT) is immediately followed by a LOGOFF command, logging is suppressed unless it was specifically requested via LIST=YES in the ERASE command or via the OPTION command.</p> <p>If SYSLST is assigned to a (user) file and has been printed out with /PRINT *SYSLST,START-SPOOL=..., a subsequent ERASE *SYSLST command logically erases only those records which have been written since the printout.</p>
*DUMMY	Designates the dummy file *DUMMY, which is considered to "always exist" and satisfies all selection criteria. All operands except the control operands (CHECK, LIST, NOSTEP) are checked for formal errors but otherwise ignored. If *DUMMY is specified, no catalog or data access takes place. *DUMMY is particularly useful for the testing of procedures.

**Keyword operands (in alphabetical order)**

ACCESS	Selection operand; the ACCESS value in the catalog entry is used as a selection criterion.
= <u>ANY</u>	The ACCESS value is not used as a selection criterion; default value.
=READ	Only files for which write access is forbidden by ACCESS=READ, i.e. read-only files, are erased.
=WRITE	All files for which write access is permitted are erased.
BACKUP	Selection operand; the files to be erased are selected on the basis of the backup level.
= <u>ANY</u>	The backup level is not used as a selection criterion; default value.
=A	Only files whose catalog entry contains BACKUP=A are erased.
=B	Only files with BACKUP=B are erased.
=C	Only files with BACKUP=C are erased.
=D	Only files with BACKUP=D are erased.
=E	Only files with BACKUP=E are erased.
BLKCTRL	Selection operand; files to be erased are selected by means of the data format. The data format is determined by the existence and position of the block control field or PAMKEY, both of which contain management information concerning the data block.
= <u>ANY</u>	The file type is not used as a selection criterion; default value.
=PAMKEY	Only files which use a separate PAM key are erased.
=DATA	Only files in which the block control field is located at the beginning of the PAM page are erased.
=NO	PAM files: only files which have no block control field are erased. ISAM and SAM files: only files in which the block control field is located at the beginning of the PAM page are erased.
=NONE	Only files for which no BLKCTRL value has been defined are erased.

CATALOG	<p>Action operand; only for files, file generations and file generation groups on private volumes. The catalog entries of the files explicitly specified or specified as partially qualified or of files selected using wildcards are erased, and their storage space is released but not overwritten. Password protection is observed, but write protection defined via ACCESS=READ or implied by RETPD is ignored; the field DESTROY in the catalog entry is also ignored.</p> <p>For tape files, "CATALOG" is the default value for the execution of ERASE.</p> <p>The action ERASE...,CATALOG is equivalent to exporting files (cf. the VOLUME operand). These files can be imported again later, either individually using FILE commands containing STATE=FOREIGN or by means of the IMPORT command, which can import one or more files on private disks at the same time. Exclusively reserved files cannot be exported.</p>
CHECK	<p>Control operand. If specified in interactive mode (or in a dialog procedure), deletion takes place in the course of the dialog. The type of dialog guidance depends on the operand value specified.</p>
=STD	<p>Is the default setting, which varies depending on the mode: CHECK=MULTIPLE is used for interactive mode, but can be changed; CHECK=NO is used for batch mode and cannot be changed.</p>
=NO	<p>The deletion procedure cannot be changed. All explicitly specified files or those specified as partially qualified or those selected using wildcards are erased.</p>
=MULTIPLE	<p>If "pathname" is partially qualified or contains wildcards, it is possible, each time the catalog ID is switched, to decide whether or not to erase files from the new catalog. (The response to the system query is either "YES" or "NO". CHECK=MULTIPLE is practical if wildcards are specified for "catid" in "pathname".) ERASE can be terminated interactively (respond with "TERMINATE" to the system query), or the CHECK mode can be changed interactively (→ NO/ERROR/SINGLE/PVS).</p>

**=ERROR** CHECK=ERROR is used to specify that a dialog as for CHECK=SINGLE is to be started if user-correctable errors occur. As long as no errors occur, CHECK=ERROR is equivalent to CHECK=NO (i.e. no dialog). CHECK=ERROR is set implicitly if CHECK=SINGLE is selected.

In the event of an error, the error message to abort ERASE processing can be acknowledged (response: "TERMINATE"), or an attempt to rectify the error can be made (response: "IGNORE="/"PASSWORD="). If desired, the CHECK mode could also be changed (→ NO/PVS/SINGLE).

**=PVS** As for CHECK=MULTIPLE, ERASE processing switches to guided dialog if files in different catalogs are affected by the ERASE command. Respond with either "YES" or "NO" to the system query, abort ERASE processing ("TERMINATE") or change the CHECK mode (→ NO/ERROR/SINGLE).

**=SINGLE** For each file which is processed, the user can decide in interactive mode whether or not it is to be erased (response YES/NO). If, in the dialog, protection attributes are specified with "IGNORE" or one or more passwords with "PASSWORD", these specifications are evaluated for the file in question, which is then erased without further queries (explicit "YES"). The user can also abort ERASE processing (response: "TERMINATE") or change the CHECK mode (→ NO/ERROR/PVS).

The affected files are listed in alphanumeric order. If file generation groups are affected, the generations are listed separately in the order of their generation numbers. If you elect not to erase a file generation, processing of the file generation group is terminated and the current status is saved (there must be no gaps in the sequence of file generations).

If only parts of a generation group are to be erased, the order of the generations depends on the value of the POS operand: for POS=AFTER, the generations are listed in descending order of their generation numbers, starting with the youngest generation; for POS=BEFORE, they are listed in ascending order of their generation numbers, starting with the oldest generation.


- CRDATE** Selection operand; the files to be erased can be selected by means of the creation date. File generation groups and file generations are not included. "CRDATE" corresponds to the catalog field "CRDATE".
- The conventions for specifying dates are described under the FSTATUS command. Range specifications include their limit values.
- =ANY** The creation date is not used as a selection criterion; default value.
- =NONE** Only files whose catalog entry field CRDATE contains the value NONE are erased, i.e. files which have never been opened.
- =date** Only files whose catalog entry field CRDATE contains precisely the specified date are erased.
- =(date[,])** Only files subjected to write access since the specified date are erased ( $\text{CRDATE} \geq \text{date}$ ).
- =(,date)** Only files which have not been updated since the specified date are erased ( $\text{CRDATE} \leq \text{date}$ ).
- =(date1,date2)** Only files which were last updated during the specified period are erased ( $\text{date1} \leq \text{CRDATE} \leq \text{date2}$ ); "date1" and "date2" must not be the same.
- DATA** Action operand; only for disk files. The data of the affected files is "logically erased". The catalog entry and the space allocation are retained. The catalog entry is identical to that for a file which has been created with FILE but not yet opened (FCBTYPE=NONE, CRDATE=NONE). In the case of tape files, the CATALOG operand goes into effect.
- DELETE-OR-EXPORT** Action operand; the effects depend on the type of volume on which the files in question are stored:
- Files, FGGs, etc. on public volumes: the catalog entry is deleted and the storage space is released (corresponds to the specification "SPACE-CATALOG").
  - Files, FGGs, etc. on private volumes: only the catalog entry is deleted (corresponds to the specification "CATALOG").

- DESTROY** Action operand; only for disk files. The storage space for the affected files is physically erased by overwriting, and then released. The catalog entry is deleted. In the case of files on private disks, all volumes on which the file was stored must be mounted at erase time.
- "Data destruction" (i.e. physical erasure) can also be selected in the CATALOG command; in this case, DESTROY=YES is set in the catalog entry.
- This causes the storage space released to be automatically overwritten. When the file is erased, first the data is overwritten and then the action operands are evaluated. If the file is to be exported (specification CATALOG or DELETE-OR-EXPORT), the data is not overwritten, since the storage space is not released. In the case of tape files, the CATALOG operand takes effect.
- EXDATE** Selection operand; files to be erased are selected by means of the expiration date. The EXDATE field in the catalog entry (cf. the FSTATUS command indicates the retention period defined for the file if the values in the fields EXDATE and CRDATE are different. File generation groups and file generations are not included.
- The conventions for specifying dates are described under the FSTATUS command. Range specifications include their limit values.
- =ANY** The expiration date is not used as a selection criterion; default value.
- =NONE** Only files whose catalog entry contains no expiration date (i.e. files which have never been opened) are erased.
- =date** Only files whose catalog entry field EXDATE contains precisely the specified date are erased.
- =(date[,])** Only files whose retention period expires on or after the specified date are erased ( $EXDATE \geq DATE$ )
- =(,date)** All files whose retention period expires on or before the specified are erased ( $EXDATE \leq date$ ).
- =(date1,date2)** Only files whose retention period expires within the specified period are erased ( $date1 \leq EXDATE \leq date2$ ); "date1" and "date2" must not be the same.

EXTENTS	Selection operand; for disk files only. The files to be processed by ERASE are selected on the basis of the number of extents. An extent is a contiguous area occupied on disk by a file. The number of extents occupied by a file is shown in the "EXTCNT" field. File generation groups and file generations are not included. "number" must be: $0 \leq \text{number} \leq 65535$ . Range specifications include their limit values.
= <u>ANY</u>	The number of extents is not used as a selection criterion; default value.
=number	Only disk files with precisely the specified number of extents are erased ( $\text{EXTCNT} = \text{number}$ ).
=(number[,])	Only disk files with at least the specified number of extents are erased ( $\text{EXTCNT} \geq \text{number}$ ).
=(number1, number2)	All disk files occupying a number of extents within the range "number1" through "number2" are erased ( $\text{number1} \leq \text{EXTCNT} \leq \text{number2}$ ); "number1" and "number2" must not be the same.
=(,number)	Only disk files with no more than the specified number of extents are processed ( $\text{EXTCNT} \leq \text{number}$ ).
FCBTYPE	Selection operand; files to be erased can be selected by means of the FCBTYPE field of the catalog entry. File generation groups and file generations are not included.
= <u>ANY</u>	The FCBTYPE is not used as a selection criterion; default value.
=NONE	Only files for which FCBTYPE=NONE applies are erased, i.e. files which have not yet been opened.
=ISAM	Only ISAM files are erased.
=BTAM	Only BTAM files are erased. These are tape files, which means that only CATALOG may be used as an action operand (the specification DELETE-OR-EXPORT is equivalent to CATALOG).
=SAM	Only SAM files are erased.
=PAM	Only PAM files are erased.

<b>FREESIZE</b>	Selection operand; only for disk files. The free PAM pages, i.e. the number of pages reserved for the file but not actually occupied, can then be used to select the files to be erased. File generation groups and file generations are not included.  "number" must be: $0 \leq \text{number} \leq 16\,777\,215$ . Range specifications are inclusive of their limit values.
<b>=<u>ANY</u></b>	The number of free PAM pages is not used as a selection criterion; default value.
<b>=SIZE</b>	Only files which occupy no storage space (i.e. files for which no PAM page has been written) are erased.
<b>=number</b>	Only files which have precisely the specified number of free PAM pages are erased.
<b>=(number[,])</b>	Only files which have at least the specified number of free PAM pages are erased ( $\text{FREESIZE} \geq \text{number}$ ).
<b>=(,number)</b>	Only files which have no more than the specified number of free PAM pages are erased ( $\text{FREESIZE} \leq \text{number}$ ).
<b>=(number1, number2)</b>	Only files which have a number of free PAM pages in the range "number1" through "number2" are erased ( $\text{number1} \leq \text{FREESIZE} \leq \text{number2}$ ); "number1" and "number2" must not be the same.
<b>IGNORE</b>	File protection operand; this specifies whether the file protection attributes <b>ACCESS=READ</b> and/or <b>EXDATE</b> and file protection with <b>BASIC-ACL</b> are to be ignored. The specification <b>IGNORE</b> in the <b>ERASE</b> command thus makes it unnecessary to issue <b>CATALOG</b> commands to reset the protection attributes before the files can be erased.  If the operand values <b>ACCESS</b> and <b>EXDATE</b> are specified in the form of a list, the two protection attributes are ignored (logical OR).
<b>=<u>NONE</u></b>	The protection attributes "read-only" ( <b>ACCESS=READ</b> ) and "retention period" ( <b>EXDATE</b> ) are observed during erasure; default value.
<b>=ACCESS</b>	Files for which <b>ACCESS=READ</b> has been defined or for which access rights have been assigned with <b>BASIC-ACL</b> can still be processed by <b>ERASE</b> .



=EXDATE	Files for which a retention period still exists (EXDATE > current date) may nevertheless be processed by ERASE.
	 At least 3 letters (EXD) must be entered for EXDATE.
LADATE	Selection operand; files to be erased can be selected by means of the last access date. File generation groups and file generations are not included.
	The conventions for specifying dates are described under the FSTATUS command. Range specifications include their limit values.
= <u>ANY</u>	The LADATE field of the catalog entry is not used as a selection criterion; default value.
=NONE	Only files whose catalog entry contains LADATE=NONE (i.e. files which have never been opened) are erased.
=date	Only files whose catalog entry field LADATE contains precisely the specified date are erased.
=(date[,])	Only files which have been accessed since the specified date (LADATE ≥ date) are erased.
=(,date)	Only files which have not been accessed since the specified date (LADATE ≤ date) are erased.
=(date1,date2)	Only files which were last accessed within the specified period are erased (date1 ≤ LADATE ≤ date2); "date1" and "date2" must not be the same.
LIST	Control operand; specifies whether successful execution of the ERASE command is to be logged to SYSOUT. In the event of error, a SYSOUT log with the name of the affected file is always generated.
= <u>NO</u>	No (additional) list with the names of the erased files is generated; default value.
=YES	A SYSOUT log containing the names of the erased files in alphanumeric order is generated.
MIGRATE	Selection operand; the files to be deleted are selected on the basis of the catalog entry in the MIGRATE field. This field indicates whether a file may be migrated from the normal public memory to a background storage level with a slower access time (see the "HSMS" manual).

<u>=ANY</u>	The files are deleted regardless of the entry; default value.
=ALLOWED	Only files which may be migrated are deleted.
=INHIBIT	Only files which may not be migrated are deleted.
MOUNT	Action operand; only for files on private volumes. This operand is used to reserve the first or all private volumes which are required. It should be specified together with the SPACE-CATALOG or DESTROY operand. In the case of tape files or files on public disks, this operand is ignored.
<u>=FIRST-DISK</u>	Only the private disk containing the beginning of the file and the catalog entry must be online; default value.
=ALL-DISK	All private disks on which sections of the file are stored must be online. If any disk is missing, the file is not erased and a SPIN-OFF is triggered after termination of ERASE processing.
NOSTEP	Control operand: in procedures, this can be used to specify whether a SPIN-OFF is to be triggered for all errors or whether -- and, if so, which -- error conditions may be ignored. If an error occurs during deletion of a system file, no SPIN-OFF is normally triggered, regardless of the setting for NOSTEP.
<u>=NONE</u>	All errors trigger a SPIN-OFF; default value.
=errcode	This can be used in conjunction with the DMS error code to specify errors which are to be ignored, i.e. which are not to cause abnormal termination of the command and to trigger a SPIN-OFF.  The DMS error code consists of 7 characters, the first three of which are always "DMS". The last four characters identify the error. The error codes specified by the user are not checked for validity as they are entered.  A list of up to three DMS error codes which are to be ignored can be specified here.
PASS	Selection operand; selects the files to be erased via the type of password.
<u>=ANY</u>	The type of password protection is not used as a selection criterion; default value.
=NONE	Only files for which no password has been defined are erased.
=EXPASS	Only files protected by an execute password are erased.

=RDPASS	Only files protected by a read password are erased.
=WRPASS	Only files protected by a write password are erased.
PASSWORD	File protection operand; specifies one or more passwords, thus permitting files protected by these passwords to be erased. The passwords entered here are not entered in the password table for the job, and are thus valid only for the current ERASE command.  The passwords must comply with the rules for password definition; in logs they are replaced by the letter "P".
= <u>NONE</u>	No passwords are specified in the ERASE command; default value.
=password	Protection by this password is to be revoked.  Specification of a password whose binary code is X'00' is equivalent to PASS=NONE. A list of up to three passwords may be specified.
POS	Selection operand; only for file generations. Wildcards may be used everywhere in "pathname" except in the generation number, which must be entered as an absolute or relative generation number. The generation identified by "pathname" must exist and is not erased.  Depending on the operand value AFTER/BEFORE, all younger or all older file generations are erased. The following fields of the catalog entry are updated: <ul style="list-style-type: none"> <li>– FIRSTGN if the oldest generation is erased; the generation specified in "pathname" becomes the oldest generation.</li> <li>– LASTGN if the youngest generation is erased; the generation specified in "pathname" becomes the youngest generation.</li> <li>– BASE if the generation with relative generation number 0 is erased; the generation specified in "pathname" becomes the base generation.</li> </ul>
=AFTER	All generations selected by "pathname" and with a generation number greater than that specified in "pathname" are erased.
=BEFORE	All generations selected by "pathname" and with a generation number less than that specified in "pathname" are erased.
SHARE	Selection operand; can be used to specify the shareability of a file as a selection criterion.
= <u>ANY</u>	Shareability is not used as a selection criterion; default value.
=NO	Only files which are not shareable, i.e. which only the user can access, are erased.

=YES	Only shareable files are erased.
=SPECIAL	Only files which can be accessed under the maintenance user ID are processed.
SIZE	<p>Selection operand; only for disk files. This is used to select files to be erased via the file size or the size of the reserved space (= number of PAM pages). File generations and file generation groups are not included.</p> <p>"number" specifies a number of PAM pages, where <math>0 \leq \text{number} \leq 16777215</math>. Range specifications include their limit values.</p>
= <u>ANY</u>	The file size is not used as a selection criterion; default value.
=FREESIZE	Only files for which space has been reserved but not actually used (LASTPG = 0) are erased, i.e. files which have not yet been opened.
=number	Only files for which precisely the specified number of PAM pages have been reserved are erased.
=(number[,])	Only files for which at least the specified number of PAM pages have been reserved are erased (SIZE $\geq$ number).
=(,number)	Only files for which no more than the specified number of PAM pages have been reserved are erased (SIZE $\leq$ number).
=(number1,number2)	All files for which a number of PAM pages in the range "number1" through "number2" have been reserved are erased.
SPACE	Action operand; only for files on public disk. The storage space for the files affected by ERASE is released, while the catalog entry is retained but updated: it is then identical to that of a file that has never been opened (FCBTYPE=NONE, CRDATE=NONE). The SPACE operand is rejected for files on private disk. In the case of tape files, the CATALOG operand goes into effect.
SPACE-CATALOG	Action operand; default value for ERASE for disk files. The catalog entries for the affected files are deleted and their storage space is released.

- STORAGE-LEVEL Selection parameter; selects files to be processed based on the storage level on which they are stored (see the "HSMS" manual).
- =ANY The storage level is not used as a selection criterion; default value.
  - =S0 Only the files on storage level S0 are erased.
  - =S1 Only the files on storage level S1 are erased. Since only the entries in the system catalog are actually deleted, the files can be restored by means of ARCHIVE or HSMS (Hierarchical Storage Management System).
  - =S2 Only the files on storage level S2 are erased. Since only the entries in the system catalog are actually deleted, the files can be restored by means of ARCHIVE or HSMS (Hierarchical Storage Management System).
- SUPPORT Selection operand; used to select the files to be erased via the volume type. File generation groups and file generations are not included.
- =ANY The volume type is not used as a selection criterion; default value.
  - =PUBLIC Only files on public disks are erased.
  - =PRDISC Only files on private disks are erased.
  - =TAPE Only files on tapes or tape cartridges are erased.
- TYPE Selection operand; specifies the file type and thus determines the extent to which ERASE selection criteria are evaluated, since not all selection operands include FGGs and file generations.
- =ANY "Normal" files, file generations and file generation groups are erased. However, FGGs and file generations are ignored by some of the selection operands in order to avoid creating gaps in the sequence of generations; default value.
  - =FILE File generation groups and file generations are not erased; all other selection operands are evaluated.
  - =FGG Only file generation groups and file generations are erased. Only selection operands which refer to attributes that are the same for all generations of an FGG (ACCESS, BACKUP, PASS, SHARE, EXDATE, SUPPORT=PRDISC) should be specified together with TYPE=FGG. All other selection operands are ignored in connection with FGGs and file generations.

- VOLUME** Selection operand; only for files on private volume. By specifying the VSN of a volume, the user can define that only files on this volume are to be erased.
- The **VOLUME** operand has a special function when specified together with the action operand **CATALOG**. If **VOLUME** indicates a private disk, individual file generations can be exported with **ERASE CATALOG, VOLUME=vs**n, which means that gaps are left in an FGG. If **VOLUME** indicates a tape or if **VOLUME** is specified without **CATALOG**, FGGs and file generations are ignored.
- If **VOLUME** is used together with the action operands **SPACE-CATALOG, SPACE, DESTROY** or **DATA**, or for tapes, FGGs and file generations are ignored.
- =ANY** Volumes are not used as a selection criterion; default value.
- =vs**n All files which are stored on the specified volume or have dummy entries on this volume are erased. In this case, "pathname" does not need to be specified.
- If **VOLUME=vs**n is specified together with the action operand **CATALOG** and **VOLUME** indicates a private disk, FGGs and file generations are also deleted.
- The user should ensure that the exporting of generations does not leave gaps in the generation sequence of an FGG.
- The combination **ERASE [pathname], CATALOG, VOLUME=vs**n is the "reverse" function to that of the **IMPORT** command (q.v.). Whereas "**ERASE, CATALOG, VOLUME**" exports all files which occupy storage space on the specified volume, the **IMPORT** command imports only those files which begin on the specified volume.

## Examples

### Example 1: Erasing files with a partially-qualified file name

```
(IN) ERASE MAX.PROTO.,CHECK=SINGLE
(OUT) % DMS0516 ERASE FILE(S) ' :W:$US123456.MAX.PROTO.CREATE-POOL' ?
      REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?
(IN) N
(OUT) % DMS0517 DELETE FUNCTION WITHDRAWN BY CALLER
      % DMS0516 ERASE FILE(S) ' :W:$US123456.MAX.PROTO.DEL-F-GROUP.1' ?
      REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?
(IN) Y
(OUT) % DMS0801 ERASE ERROR ON FILE :W:$US123456.MAX.PROTO.DEL-F-GROUP.1
      % DMS05BF FILE PASSWORD-PROTECTED. FIRST ENTER CORRECT PASSWORD WITH
      APPROPRIATE COMMAND, THEN REENTER /DELETE-FILE OR /ERASE
      % DMS0516 ERASE FILE(S) ' :W:$US123456.MAX.PROTO.DEL-F-GROUP.1' ?
      REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?
(IN) ?
      % DMS0516 ERASE FILE(S) ' (&00)' ? REPLY (Y=YES; N=NO; T=TERMINATE
      CMD; ?=EXPL. OF ADDITIONAL OPTIONS)
      % - The user can decide, in dialog mode, whether the erase task requested
      % - is to be processed or not.
      % - (&00): Partially qualified file name, file name, name of file
      % - generation or file generation group.
      % - Y: The file or file generation or file generation group will be
      % - erased.
      % - N: This at times displayed file or file generation or file
      % - generation group will not be erased.
      % - T: The erase command entered will not be processed any further.
      % - ?: The explanation of the additional options will be displayed following
      % - the request which will be offered once more.
      % - The meaning of the operands of the additional options permitted
      % - is analogous to that of the corresponding operands in the command
      % - and may be looked up in the reference manual 'Control System Command
      % - language'.
      % - Additional options which may be added with comma to the reply
      % - ('Y' or 'N' or 'T'):
      % - ,CHECK: Update of the current dialog mode:
      % - ,CHECK=NO or =PVS or =MULTIPLE or =SINGLE or =ERROR.
      % - ,IGNORE: Re-issue of the protection attribute before the erase job is
      % - processed:
      % - ,IGNORE=ACCESS or =EXDATE or ,IGNORE=(ACCESS,EXDATE).
      % - ,PASSWORD: Re-issued of passwords before erase job is processed.
      % - Only maximum of 3 passwords is allowed hexadecimally or
      % - numeric.
      % - ,PASSWORD=password or =(password,...).
      % - Enter reply corresponding to the explanation in the meaning text.
      % - If the reply is invalid, or if an empty string is entered, the task
      % - will not be processed.
      % DMS0516 ERASE FILE(S) ' :W:$US123456.MAX.PROTO.DEL-F-GROUP.1' ?
      REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?
(IN) N
(OUT) % DMS0517 DELETE FUNCTION WITHDRAWN BY CALLER
      % DMS0516 ERASE FILE(S) ' :W:$US123456.MAX.PROTO.DEL-F-GROUP.21' ?
      REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?
(IN) Y
(OUT) % DMS0516 ERASE FILE(S) ' :W:$US123456.MAX.PROTO.ERASE.1' ?
      REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?
(IN) Y
(OUT) % DMS0801 ERASE ERROR ON FILE :W:$US123456.MAX.PROTO.ERASE.1
      % DMS06D5 THE FILE IS READ ONLY ACCESS. COMMAND TERMINATED
      % DMS0516 ERASE FILE(S) ' :W:$US123456.MAX.PROTO.ERASE.1' ?
      REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?
(IN) Y,IGNORE=ACCESS
(OUT) % DMS0516 ERASE FILE(S) ' :W:$US123456.MAX.PROTO.ERASE.2' ?
      REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?
(IN) Y
(OUT) % DMS0801 ERASE ERROR ON FILE :W:$US123456.MAX.PROTO.ERASE.2
      % DMS05C3 FILE TO BE DELETED IS IN USE. RETRY COMMAND LATER
(IN) FSTATUS MAX.PROTO.
(OUT) 00000000 :W:$US123456.MAX.PROTO.CREATE-POOL
      00000024 :W:$US123456.MAX.PROTO.DEL-F-GROUP.1
```

## ERASE

---

```
00000024 :W:$US123456.MAX.PROTO.ERASE.2
:W: PUBLIC: 3 FILES. RES= 48, FREE= 35, REL= 33 PAGES
(IN) ERASE MAX.PROTO.DEL-F-GROUP.1, PASSWORD=P
(IN) FSTAT MAX.PROTO.
(OUT) 00000000 :W:$US123456.MAX.PROTO.CREATE-POOL
00000024 :W:$US123456.MAX.PROTO.ERASE.2
:W: PUBLIC: 2 FILES. RES= 24, FREE= 24, REL= 24 PAGES
```

A number of files which were created as logging files are to be erased (subname: PROTO). The file MAX.PROTO.CREATE.POOL is to be retained. The file MAX.PROTO.DEL-F-GROUP.21 is password-protected, the file MAX.PROTO.ERASE.1 has write protection, MAX.PROTO.ERASE.2 is locked. User prompting is requested during execution.

Finally, the file MAX.PROTO.DEL-F-GROUP.21 is erased with the PASSWORD operand.



*Example 2: Wildcard \**

```
(IN)      FSTAT ,CRDAT=+0
(OUT)    00000003 :W:$US123456.MAX.TEST.1
          00000003 :W:$US123456.MAX.TEST.2
          :W: PUBLIC:      2 FILES. RES=      6, FREE=      4, REL=      0 PAGES

(IN)      ERASE **,CRDAT=+0,LIST=YES
(OUT)    % DMS0516 ERASE FILE(S) ':W:$US123456.**' ? REPLY (Y=YES; N=NO;
          T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?

(IN)      Y
(OUT)    % DMS0800 SPECIFIED FILE ':W:$US123456.MAX.TEST.1' ERASED
(OUT)    % DMS0800 SPECIFIED FILE ':W:$US123456.MAX.TEST.2' ERASED
```

/ERASE \*\* erases all files under the user ID ( $\neq$  FSTAT \*); the selection is in this case restricted by the CRDATE operand (ERASE \* would erase the object module file of the job).

/ERASE \*SYSLST would erase system file SYSLST, /ERASE \*\*SYSLST would erase all files whose file name ends with the character string SYSLST.

*Example 3: Selection operands*

```
(IN)      FS MAX.*ASS.
(OUT)    00000000 :W:$US123456.MAX.Q.ASS.DSECTS
          00000000 :W:$US123456.MAX.Q.ASS.DSECTS3
          00000003 :W:$US123456.MAX.T1ASS.FILE
          :W: PUBLIC:      3 FILES. RES=      3, FREE=      2, REL=      0 PAGES

(IN)      ERASE *ASS.,SIZE=(0, ),LIST=YES,CHECK=SINGLE
(OUT)    % DMS0516 ERASE FILE(S) ':W:$US123456.MAX.Q.ASS.DSECTS' ?
          REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?

(IN)      Y
(OUT)    % DMS0800 SPECIFIED FILE ':W:$US123456.MAX.Q.ASS.DSECTS' ERASED
          % DMS0516 ERASE FILE(S) ':W:$US123456.MAX.Q.ASS.DSECTS3' ?
          REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?

(IN)      N
(OUT)    % DMS0517 DELETE FUNCTION WITHDRAWN BY CALLER
          % DMS0516 ERASE FILE(S) ':W:$US123456.MAX.T1ASS.FILE' ?
          REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?

(IN)      Y
(OUT)    % DMS0800 SPECIFIED FILE ':W:$US123456.MAX.T1ASS.FILE' ERASED
(IN)      FSTAT MAX.
(OUT)    00000003 :W:$US123456.MAX.MESS.SHORT
          00000024 :W:$US123456.MAX.PROTO.ERASE.2
          00000000 :W:$US123456.MAX.Q.ASS.DSECTS3
          00000000 :W:$US123456.MAX.TEST.GROUP.1 (FGG)
          :W: PUBLIC:      4 FILES. RES=      27, FREE=      20, REL=      15 PAGES
```

*Example 4: Deleting file generations*

```
(IN) FSTATUS MAX. ,TYPE=FGG,GEN=YES
(OUT) 00000000 :W:$US123456.MAX.GROUP.RES (FGG)
      00000003 :W:$US123456.MAX.GROUP.RES(*0002)
      00000003 :W:$US123456.MAX.GROUP.RES(*0003)
      00000003 :W:$US123456.MAX.GROUP.RES(*0004)
      00000003 :W:$US123456.MAX.GROUP.RES(*0005)
      00000003 :W:$US123456.MAX.GROUP.RES(*0006)
      00000000 :W:$US123456.MAX.TEST.GROUP.1 (FGG)
      00000003 :W:$US123456.MAX.TEST.GROUP.1(*0009)
      :W: PUBLIC: 8 FILES. RES= 18, FREE= 12, REL= 0 PAGES

(IN) ERASE MAX.*GR*( *4) ,TYPE=FGG,POS=AFTER,LIST=YES
(OUT) % DMS0516 ERASE FILE(S) ':W:$US123456.MAX.*GR*( *0004)' ?
      REPLY (Y=YES; N=NO; T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?

(IN) Y
(OUT) % DMS0800 SPECIFIED FILE ':W:$US123456.MAX.GROUP.RES(*0006)' ERASED
      % DMS0800 SPECIFIED FILE ':W:$US123456.MAX.GROUP.RES(*0005)' ERASED
      % DMS0801 ERASE ERROR ON FILE :W:$US123456.MAX.TEST.GROUP.1
      % DMS06C7 USER ATTEMPTED TO PROCESS A GENERATION WITH AN INCORRECT
      GENERATION NUMBER. COMMAND TERMINATED
```

*Example 5: Exporting files to private disk*

```

(IN)      FSTAT ,SUPPORT=PRDISC
(OUT)    00000003*:W:$US123456.MAX.PRIV.DAT.1
          00000003*:W:$US123456.MAX.PRIV.DAT.2
          00000003*:W:$US123456.MAX.PRIV.DAT.3
          00000003*:W:$US123456.MAX.PRIV.DAT.4
          00000003*:W:$US123456.MAX.PRIV.DAT.5
:W:  PRIVATE:    5 FILES. RES=    15, FREE=    10, REL=    0 PAGES

(IN)      ERASE ,DELETE-OR-EXPORT,VOLUME=WORK02
(OUT)    % DMS0516 ERASE FILE(S) ' :W:$US123456.' ? REPLY (Y=YES; N=NO;
          T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?

(IN)      N
(OUT)    % DMS0517 DELETE FUNCTION WITHDRAWN BY CALLER

(IN)      ERASE MAX. ,DELETE-OR-EXPORT,VOLUME=WORK01
(OUT)    % DMS0516 ERASE FILE(S) ' :W:$US123456.MAX.' ? REPLY (Y=YES; N=NO;
          T=TERMINATE CMD; ?=EXPL. OF ADDITIONAL OPTIONS)?

(IN)      Y

(IN)      FSTAT ,SUPPORT=PRDISC
(OUT)    00000003*:W:$US123456.MAX.PRIV.DAT.5
:W:  PRIVATE:    1 FILE. RES=    3, FREE=    2, REL=    0 PAGES

(IN)      IMPORT ,VOLUME=WORK01,DEVICE=D3480,LIST=(YES,SYSOUT)
(OUT)    0 :W:$US123456.MAX.PRIV.DAT.1
          0 :W:$US123456.MAX.PRIV.DAT.2
          0 :W:$US123456.MAX.PRIV.DAT.3
          0 :W:$US123456.MAX.PRIV.DAT.4

(IN)      FSTAT ,SUPPORT=PRDISC
(OUT)    00000003*:W:$US123456.MAX.PRIV.DAT.1
          00000003*:W:$US123456.MAX.PRIV.DAT.2
          00000003*:W:$US123456.MAX.PRIV.DAT.3
          00000003*:W:$US123456.MAX.PRIV.DAT.4
          00000003*:W:$US123456.MAX.PRIV.DAT.5
:W:  PRIVATE:    5 FILES. RES=    15, FREE=    10, REL=    0 PAGES

```

## ESCAPE Interrupt procedure

Application group: Job control (page 22 ff.)

### Command description

The ESCAPE command has the same effect as the ESCAPE function (pressing K2 key). A procedure is interrupted at the point where an ESCAPE command is encountered, and a message is displayed on the screen, giving the level number of the interrupted procedure:

TASK IS IN ESCAPE MODE AT LEVEL NUMBER I

I = 0 indicates level 0 (primary command input)

I = 1 indicates level 1

I = 2 indicates level 2

etc.

You can subsequently enter commands from the terminal. ESCAPE mode remains in force until left by means of the RTI or the ABORT command.

This command is only valid in interactive mode and can only be used in a procedure. It is ignored in batch mode.

Nesting of interrupts is permissible (see examples). A procedure called in ESCAPE mode can also be interrupted by means of an ESCAPE command or ESCAPE function. In this case, the first interrupt is not lost (see example 4).

In ESCAPE mode, the symbolic operands which were defined in the interrupted procedure are accessible to users, and may be entered from the terminal. However, a command with a symbolic operand which has been replaced by the appropriate value is only listed if a log for the interrupted procedure was requested (see example 5).

### Format

Operation	Operands
ESCAPE	

The command is entered without operands.

## Examples

*Example 1:* Execution of an ENDP command in ESCAPE mode

Fig. ESCAPE-1 ENDP command in ESCAPE mode

### Assignment of SYSCMD

- 1) Command `"/CALL PROC1"` assigns SYSCMD to the procedure file PROC1.
- 2) Command `"/CALL PROC2"` assigns SYSCMD to the procedure file PROC2.
- 3) After ESCAPE processing has been initiated, SYSCMD is assigned to the terminal.
- 4) SYSCMD is still on the data display terminal (ESCAPE mode retained). A branch is made from procedure level 2 to procedure level 1.
- 5) The RTI command terminates ESCAPE mode. SYSCMD returns to the procedure file PROC1.
- 6) SYSCMD is reassigned to primary command input.

## *Example 2: Execution of a CALL command in ESCAPE mode*

Fig. ESCAPE-2 CALL command in ESCAPE mode

### Assignment of SYSCMD

- 1) Command `"/CALL PROC1"` assigns SYSCMD to the procedure file PROC1.
- 2) After ESCAPE processing has been initiated, SYSCMD is assigned to the data display terminal.
- 3) Command `"/CALL PROC2"` assigns SYSCMD to the procedure file PROC2.
- 4) SYSCMD is still on the data display terminal (ESCAPE mode retained). A branch is made from procedure level 2 to procedure level 1.
- 5) The RTI command terminates ESCAPE mode. SYSCMD returns to the procedure file PROC1.
- 6) SYSCMD is reassigned to primary command input.

*Example 3:* Execution of a DO command in ESCAPE mode

Fig. ESCAPE-3 DO command in ESCAPE mode

Assignment of SYSCMD

- 1) Command `"/CALL PROC1"` assigns SYSCMD to the procedure file PROC1.
- 2) Command `"/CALL PROC2"` assigns SYSCMD to the procedure file PROC2.
- 3) After ESCAPE processing, SYSCMD is assigned to the data display terminal.
- 4) Command `"/DO PROC3"` assigns SYSCMD to the procedure file PROC3, but there is no further nesting (level 2 is retained).
- 5) SYSCMD is returned to the data display terminal (ESCAPE mode is not terminated). A branch is made from procedure level 2 to procedure level 1.
- 6) The RTI command terminates ESCAPE mode. SYSCMD is once more on the procedure file PROC1.
- 7) SYSCMD is reassigned to primary command input.

## *Example 4:* Nesting of ESCAPE levels

Fig. ESCAPE-4 Nesting of ESCAPE levels

### Assignment of SYSCMD

- 1) Command `"/CALL PROC1"` assigns SYSCMD to the procedure file PROC1.
- 2) SYSCMD is then assigned to the data display terminal, after ESCAPE processing has been initiated.
- 3) Command `"/CALL PROC2"` assigns SYSCMD to the procedure file PROC2.
- 4) SYSCMD is then assigned to the data display terminal, after ESCAPE processing has been initiated.
- 5) SYSCMD is assigned to the procedure file PROC2.
- 6) SYSCMD is assigned to the data display terminal (ESCAPE mode for 1st level still in force).
- 7) Command RTI assigns SYSCMD to the procedure file PROC1.
- 8) SYSCMD is returned to primary command input.



*Example 5:* Use of symbolic parameters in ESCAPE mode

Fig. ESCAPE-5 Symbolic parameters in ESCAPE mode

The arrows show the path of the system file SYSCMD. When the PROC1 procedure is called by the CALL command, the file name "TEST" is entered for the operand "&DAT". After exiting procedure mode by means of the ESCAPE command, the user issues the SYSFILE command at the terminal, using the symbolic parameter "&DAT". Since command logging to SYSOUT was requested for the PROC1 procedure file (operand C specified in the PROCEDURE command), this command will be logged to SYSOUT even in ESCAPE mode, complete with the appropriate value for TEST.

## EXECUTE      Load and start program

Application group: Program control (page 30)

### Command description

The EXECUTE command

- links object modules and link load modules, reads them to memory and initiates them (dynamic linking loader (DLL) call; format 1),
- reads load modules to memory and initiates them (static linking loader (ELDE) call; format 2).

If the program is to be loaded but not yet initiated, use the LOAD command.

An **object module** (linkable module) is produced when a source program is compiled. An object module is stored:

- temporarily in the \* file (temporary EAM object module file),
- permanently in an object module library, OML, (library routines LMR/LMS),
- permanently as an R-type member in a program library, PL, (library routine LMS).

**Load module** is produced when one or more object modules are linked to form an executable program. A load module is stored:

- as a cataloged file (output file of the linkage editor TSOSLNK),
- as a C-type member in a program library.

A **link load module** (LLM) comprises a number of modules linked by the LINKAGE EDITOR. It has the attributes of both prelinked modules (large modules) and programs (load modules) generated by the TSOSLNK linkage editor, which results in optimum loading time. For further details refer to the "Linkage Editor and Loaders" manual [22].

The Dynamic Linking Loader (DLL) links object modules into a load module, reads this module into memory and initiates the program run. The ELDE loader reads a load module into memory and initiates the program run.

The system messages providing information about the load operation (BLS0500, BLS0517, ...) can be suppressed by activating job switch 4. For a detailed description, see the "Linkage Editor and Loaders" manual [22] and the "LMS" manual [14].

The execution of programs cannot be nested.

If an EXECUTE command is issued while a program is loaded, the loaded program is unloaded and the EXECUTE command is executed. Although no error flag is displayed an ABEND-STXIT routine defined in the unloaded program is activated, unless the EXEC command was called via the CMD macro (see the "Executive Macros" manual [5]).

**Program monitoring**

When the EXECUTE command is issued, the status indicator in the program-monitoring job variable is set to "\$R".

If a program has already been loaded (and perhaps interrupted by the BREAK command), the status indicator in the job variable monitoring this program is set to "\$A".

If the job variable is not accessible at command execution time, an error message is output to SYSOUT and command execution continued.

## Formats and operand description (identical to those of LOAD command)

### Format 1: Calling the Dynamic Linking Loader DLL

Operation	Operands
EXECUTE	<pre> * {   (entry[, { *LINK ([LINK=]linkname) }])   {     { *STD }     { *LINK ([LINK=]linkname) }     {version} [, {pathname} ]   } }  [, TIME=t] [, MONJV=jvname]  {   IDA={ YES }   { NO } }  [, {   SYMTTEST={ ALL }   { NO } }]  [, PROG-MOD={ 24 } { ANY }]  [, RUN-MOD={ STD { ADVANCED ( [LOAD-INF= { DEFINITIONS MAP NONE REFERENCES } ] [, ALTLIB={ NO } { YES } } }  [, NAME-COL={ STD { ABORT } } [, UNRES={ STD { DELAY } { ABORT } } [, ERREXIT=address]  [, CTRL-MSG={ INFORMATION { WARNING ERROR } } [MAP={ NO number SYSOUT BOTH [ (number) ] } ] [, SHARE={ SYSTEM { NONE } } ]  [, VSPACE= (a [, b]) ]  [, VPWAIT={ TASK-STD { number } } </pre>

*	The asterisk designates the task's temporary EAM object module file. The modules in this file are linked to form a program. If the object module library contains a number of modules of the same name, only the first module is loaded.
entry	Name of an entry point (ENTRY) or CSECT statement in a program. The object module with "entry" is loaded and executed from the point of entry, provided no operand is specified in the END statement. If the END statement contains an operand, its address <u>always</u> serves as the start address.
module	Name of a module (object module or LLM) or library member to be loaded by DLL. "module" may be up to 32 characters in length.
pathname	stands for: [:catid:][userid.]modulelib
catid	Catalog ID of the pubset on which the file is stored. Default value: the catalog ID assigned to the user ID (JOIN entry).
userid	User ID to which the file is assigned. Default value: the user ID from the LOGON command.
modulelib	Name of the library containing the specified object module. This library may be an object module library (OML) or a program library with members of type R and L. If "libname" and "linkname" are not specified with RUN-MOD=ADVANCED, the library assigned to link name BLSLIB is searched.  If "libname" is not specified with RUN-MOD=STD, the TASKLIB library is searched by the operating system. It is not possible to process an LLM with RUN-MOD=STD.  If the library (file) is not cataloged under the user's ID, it must be shareable (operand SHARE=YES in the CATALOG command), and the user ID must be specified.  If the "pathname" entry is missing or the required object module is not found there, it is searched for in the following files in the specified sequence: <ol style="list-style-type: none"> <li>1. \$TSOS.modlib or \$TSOS.library;</li> <li>2. File assigned to the (system) file TASKLIB (/SYSEFILE TASKLIB=...);</li> <li>3. (User) file with the name TASKLIB;</li> <li>4. \$TSOS.TASKLIB. (no search through this file if there is a (user) file with the name TASKLIB).</li> </ol>

- IDA** Allows the use of symbolic addresses for debugging programs with the aid of IDA commands, provided that an internal symbol dictionary exists for the program at that time.
- You can have this internal symbol dictionary created by entering the corresponding operand in the COMOPT statement; e.g. `**COMOPT ISD` for the assembler.
- =YES** The internal symbol dictionary is used.
- =NO** The internal symbol dictionary is ignored; default value
- LINK=linkname** Specifies the link name assigned to a library. LINK=linkname is only permitted for RUN-MOD=ADVANCED and must be specified with \*LINK(..) instead of a library name.
- MONJV** Specifies a job variable, which serves to monitor the program run. The operating system enters the program's processing state in the job variable (\$R, \$T, \$A  $\triangle$  job initiated, job terminated normally, job terminated abnormally). The STATUS MONJV=... command, for example, could then be used to check this state. Note that the job variable must be cataloged before calling this command. For information about job variables please refer to the "Job Variables" manual [11]. Job variables cannot be used unless the JV software product is installed.
- =jvname** Name of the job variable.
- PROG-MOD** Of significance only on XS31 systems. This operand is ignored on non-XS31 systems. PROG-MOD specifies the area of class 6 memory (above or below the 16-Mb boundary) into which the program is loaded. The correlation between the addressing mode (AMODE) and load mode (RMODE) specified in the program must be taken into account. (AMODE and RMODE denote statements in an assembler program.)
- =24** The module is loaded below the 16-Mb boundary. The program is executed in 24-bit addressing mode. External references are interpreted as 24-bit addresses. Loading of a program with the attribute AMODE=31 is aborted with an error message; this also applies to the loading of a load unit if the control section has the attribute AMODE=31. 24 is the default value.

=ANY The load address is established as follows by combining the addressing and load modes specified in the program:

RMODE	AMODE		
	24	31	ANY
24	≤ 16-Mb boundary	≤ 16-Mb boundary	≤ 16-Mb boundary
ANY	Error	> 16-Mb boundary	> 16-Mb boundary

Where:

≤ 16-Mb boundary: below the 16-Mb boundary.

> 16-Mb boundary: above the 16-Mb boundary.

Error: illegal combination; error message during program compilation.

RUN-MOD Defines the operating mode of the Dynamic Linking Loader DLL.

=STD The DLL operates in a mode that is fully compatible with previous BS2000 versions (DLL of BS2000 versions prior to version 10.0A).

=ADVANCED(..) DLL operates in a mode that supports the new functions (as of BS2000 V10.0A). These new functions may produce incompatibilities in the case of existing modules.

ALTLIB=YES/NO

Specifies whether alternative libraries are to be searched. Alternative libraries are assigned via the link name BLSnn (00 ≤ nn ≤ 99). They are also used for the autolink function of DLL.

NAME-COL Defines how name conflicts are to be handled in the case of identically named symbols. Name conflicts are only detected if the symbols are *not* masked.

=STD Name conflicts between unmasked symbols are indicated by means of warnings. The module containing the symbol of the same name is loaded. The new version of the symbol is masked, i.e. is no longer used to satisfy external references.

=ABORT Loading of the current load unit is aborted if a name collision between unmasked symbols is detected.

- UNRES** Defines how unresolved external references are to be treated. All unresolved external symbols are output to the symbol file SYSOUT. Unresolved external dummy sections (XDSECS-R) are listed separately.
- =STD** Unresolved external references receive an address specified in the ERROR-EXIT operand.
- =DELAY** Unresolved external references are satisfied later. This operand is only permitted for LOAD-INFORMATION=REFERENCES. DLL stores the unresolved external references in the link context. If the next load unit is loaded in this context, DLL will then attempt to satisfy the stored external references with the CSECTs and ENTRYs of this load unit. This process is repeated when further load units are loaded, as long as the context exists. External dummy sections (XDSECS-R) cannot be stored. When unresolved external references are stored in the link context, they are assigned a (provisional) address that is specified in the ERROR-EXIT operand.
- =ABORT** Unresolved external references are not permitted. Loading of the current load unit is aborted.
- ERREXIT=address** Defines the address that is assigned to the unresolved external references if the operands UNRESOLVED-EXTRNS=STD / DELAY are specified. The value X'FFFFFFFF' is used by default.
- CTRL-MSG** Defines the lowest message class as of which messages are to be output.
- =INFORMATION** Messages of all message classes are output.
- =WARNING** Only messages of message class WARNING and ERROR are output. Messages of the INFORMATION class are suppressed.
- =ERROR** Only messages of the message class ERROR are output.
- LOAD-INF** Defines the load information of the load unit.
- =DEFINITIONS** Loads an ESD containing the program definitions of all modules of the load unit. Program definitions are control sections (CSECTs), entry points (ENTRYs), COMMON areas, and external dummy sections (XDSECS-D).




- =REFERENCES** Loads an ESD containing not only the program definitions but also the resolved references of all modules of the load unit. References are external references (EXTRNs), V-type constants, conditional external references (WXTRNs), and external dummy sections (XDSECs-R).
- =MAP** Only one ESD, which is required to create the DLL listing, is loaded *temporarily*. The ESD is unloaded as soon as the DLL listing has been generated.
- =NONE** No ESD is loaded.
- MAP** Specifies whether or not a DLL listing is to be output and defines the output destination.
- =NO** No DLL listing is output.
- =num** The output destination is a system file SYSLST from the set SYSLST01 through SYSLST99 (where  $1 \leq \text{num} \leq 99$ ).
- =SYSOUT** The output destination is the system file SYSOUT.
- =BOTH[(num)]** Defines the output destination as the system file SYSOUT *and* SYSLST. "num" specifies the number of the system file SYSLST from the set SYSLST01 through SYSLST99.
- SHARE** Specifies whether or not an unprivileged subsystem (see the "System Installation" manual) and share programs are to be considered by DLL during searching. This also applies to the autolink function of DLL.
- =SYSTEM** DLL first searches in an unprivileged subsystem and then in share programs.
- =NONE** Any unprivileged subsystem and share programs are skipped by DLL during a search.
- SYMTEST** This operand is required for debugging with AID.
- =ALL** SYMTEST=ALL allows the use of symbolic addresses for program debugging with AID commands. This requires that a local or internal symbol dictionary be generated at program compilation time.

<u>=NO</u>	The symbolic address table or internal symbol dictionary is ignored; default value.
TIME	Defines a maximum CPU time for the program run. The resulting behavior is different in interactive and batch mode: <ul style="list-style-type: none"><li>– Interactive mode: After the specified time has expired, the program is interrupted and a user prompt asks whether you require a program dump. If no program dump is required, the user can opt to continue the program run, or abort. The time specified with TIME=... is not compared with that of the LOGON command. If TIME=... is not specified, the program run is not interrupted - irrespective of the time specification, if any, in the LOGON command.</li><li>– Batch mode: The program run is aborted once the specified time has expired. The program is not loaded if the specified time exceeds the CPU time still available to the task at that moment (ENTER-JOB-/LOGON command/default value). This latter specification is the maximum program run time (CPU time) unless otherwise defined by means of TIME; once this time has expired, the program run is aborted and the task terminated.</li></ul>
=t	Time specification in seconds; $0 \leq t \leq 32767$ .  If the operand TIME=NTL has been specified in the LOGON or ENTER command, the "t" entry is ignored.
VPSPACE=(a[,b])	Only permitted for vector-based computers; specifies the required vector storage space, either above ("a") or below ("b") the 16 Mb boundary. "a" and "b" are integers.
VPWAIT	Only permitted for vector-based computers; defines the maximum waiting time (in minutes) that may be used by the job for the reservation of storage space. The command is rejected when the waiting time has expired.
<u>=TASK-STD</u>	The command is rejected in interactive mode if the reservation of vector storage space is not possible. There is no restriction on the waiting time in batch mode.
=num	Specifies the waiting time in numbers of seconds.

**Format 2: Calling the static loader ELDE**

Operation	Operands
EXECUTE	<pre> {pathname   LIBRARY=pathname, ELEMENT=member [ (VERSION={version}                                      {*_STD}) ] } [ , CLASSII=(max[ , min[ , p] ) ] [ , TIME=t ] [ , {   IDA={YES       NO     }   SYMTTEST={ALL             NO             } } ] [ , MONJV=jvname ] [ , VSPACE=(a [ , b ] ) ] [ , VPWAIT={TASK-STD             num             } ] </pre>

**Operand description**

pathname	stands for: [ :catid: ] [ \$userid. ] filename
catid	Catalog ID of the pubset on which the file is stored. Default value: the catalog ID assigned to the user ID (JOIN entry).
userid	User ID to which the file is assigned. Default value: user ID from the LOGON command.
filename	Name of the cataloged file where the load module is located.
	 Names of file generations or file generations groups must <u>not</u> be specified here. If the file is not cataloged under the user's ID, it must be shareable (SHARE=YES operand in CAT command). The user ID must be specified.

- CLASSII** Concerns requests for additional memory pages for a program. A check is made as to whether the total number of pages requested (including program space) exceeds the class 6 memory space available to you. In addition, the number of required pages resident in memory must not exceed the value allowed for current operation (see the STATUS command, BIAS operand). The actual request for additionally required pages must be made in the program using the REQM or REQMP macro; for resident memory pages, the CSTAT macro must also be specified (for macros, see the "Executive Macros" manual [5]). The CLASSII operand is only effective in conjunction with the "pathname1" operand.
- =(max,min,p)** "max", "min", "p" are the number of resident and pageable memory pages required (in addition to the program size).
- "max" is the number of additional resident memory pages essential for optimum program execution.  
If these pages are not available, the value of "min" will be used for "max".
- "min" is the number of additional resident memory pages essential for program execution.  
If these pages are not available, the program will not be loaded. In batch mode, the task is assigned to the queue for memory saturation (task type T1/HO; see the STATUS command).
- "p" is the number of pageable memory pages required in addition to the program size.
- ELEMENT** Identifies a C-type member in the PL. The member identifier consists of
- the name of the member
  - version
- Specification of the version is optional. For a detailed description, see the "LMS" manual [14].

- IDA** Allows the use of symbolic addresses for debugging programs with the aid of IDA commands, provided that an internal symbol dictionary exists for the program at that time.
- You can have this internal symbol dictionary created by entering the corresponding operand in the COMOPT statement; e.g. `**COMOPT ISD` for the assembler.
- When calling the static loader ELDE, you must initiate the linking of the internal symbol dictionary by specifying `IDA=Y` for linking with `TSOSLNK` (see the "Linkage Editor and Loaders" manual [22]).
- =YES** The internal symbol dictionary is used.
- =NO** The internal symbol dictionary is ignored; default value
- LIBRARY** Specifies a program library (PL) containing the load  
**=pathname** module as a C-type member.
- MONJV** Specifies a job variable which serves to monitor the program run. The operating system enters the program's processing state in the job variable (`$R`, `$T`, `$A`  $\triangleq$  job initiated, job terminated normally, job terminated abnormally). The `STATUS MONJV=...` command, for example, could then be used to check this state. Note that the job variable must be cataloged before `/EXECUTE ...` is issued. For information about job variables please refer to the "Job Variables" manual [11]. Job variables cannot be used unless the JV software product is installed.
- =jvname** Name of the job variable.
- SYMTEST** The operand is required for debugging with AID.
- =ALL** `SYMTEST=ALL` allows the use of symbolic addresses for debugging programs with AID commands. This requires that a local or internal symbol dictionary be generated at program compilation time.
- =NO** The symbolic address table or internal symbol dictionary is ignored; default value.

TIME	Defines a maximum CPU time for the program run. The resulting behavior is different in interactive and batch mode: <ul style="list-style-type: none"><li>– Interactive mode: After the specified time has expired, the program is interrupted and a user prompt asks whether you require a program dump. If you do not require a program dump, you can continue the program run or abort. The time specified with TIME=... is not compared with that of the LOGON command. If TIME=... is not specified, the program run is not interrupted - irrespective of the time specification, if any, in the LOGON command.</li><li>– Batch mode: The program run is aborted once the specified time has expired. The program is not loaded if the specified time exceeds the CPU time still available for the task at that moment (ENTER-JOB-/LOGON command/default value). This latter specification is the maximum program run time (CPU time) unless otherwise defined by means of TIME; once this time has expired, the program run is aborted and the task terminated.</li></ul>
=t	Time specification is seconds; $0 \leq t \leq 32767$ .
VERSION	Version entry (supplements the member name).
=version	Version identifier (up to 24 characters).
=*STD	Latest version; default value.
VPSPACE=(a[,b])	Only permitted for vector-based computers; specifies the required vector storage space above ("a") or below ("b") the 16 Mb boundary. "a" and "b" are integers.
VPWAIT	Only permitted for vector-based computers; defines the maximum waiting time (in minutes) that may be used by the job for the reservation of storage space.
= <u>TASK-STD</u>	The command is rejected in interactive mode if the reservation of vector storage space is not possible. There is no restriction on the waiting time in batch mode.
=num	Specifies the waiting time in minutes.

## Examples

### *Example 1*

Fig. EXEC-1 Load and start program

*Example 2*

If the EXECUTE command is used to load and start load modules which require resident space, operand CLASSII must be specified. In interactive mode, the user can use the "STATUS BIAS" command to check in advance whether the system has sufficient space for resident programs (CORE is displayed).

```
/EXECUTE PROG.RES.2,CLASSII=(5,2,10) _____ (01)
```

(01) The program PROG-RES.2 requires a total of 10 virtual pages, at least 2 of which must be resident. Up to 5 pages can be made resident.

*Example 3*

```
(IN) EXEC $EDOR,TIME=1
(OUT)
(OUT) E D O R : FILE-EDITOR (V8.3D22)
(OUT)
      :
      :
(OUT) % IDA0N51 PROGRAM INTERRUPT AT LOCATION '00F3CA08 (EDOR), (CDUMP), EC=80'
(OUT) % IDA0N45 DUMP DESIRED? REPLY (Y=USER-/AREADUMP;Y,SYSTEM=SYSTEMDUMP; N=NO)?
(IN) N
(OUT) % EXC0075 TIME LIMIT FOR PROGRAM RUN EXCEEDED. PROGRAM TO BE CONTINUED? REPLY (Y=YES; N=NO)?
(IN) N
```

After 1 second of CPU time has elapsed (operand "TIME=1") the program is interrupted. If no dump is requested and the user responds "Y" to the inquiry regarding program continuation, the program will resume.

For further examples, see LOAD, PARAMETER and SYSFILE commands.



## **FILE Define file attributes / control file processing**

Application group: File processing (page 26 ff.)

### **Command description**

The FILE command processes permanent and temporary files (but not EAM files) and file generations. It can be used to create new files and catalog entries, to change file attributes, and to import files from private volumes.

The following pages illustrate the various features of the FILE command with the help of a summary table and a diagram.

Unlike the CATALOG command, the FILE command cannot be used to define or change file attributes such as passwords or the access type (with the exception of the retention period). If a catalog entry is created using FILE, the system default values are used for these attributes. They can then be changed by means of a CATALOG command.

The FILE command establishes a connection between the program and the file via the task file table (TFT, see below).

### **Processing the catalog entry**

If the file or file generation specified in the FILE command is not yet cataloged, a catalog entry is created under the user ID of the current job. The values entered for the operands DEVICE, VOLUME, SPACE, DDEVICE, DVOLUME, DSPACE and FSEQ (if applicable) are evaluated and transferred to the catalog entry; the values specified for other operands are only evaluated and transferred to the TFT entry if a file link name (see below) is specified by means of the LINK operand (exception: MOUNT, TVSN and STATE=FOREIGN).

If a file/file generation which is to be cataloged is stored on private disk, DMS takes the values for the catalog entry from the F1 label of the first volume containing the file.

### Task File Table (TFT)

The Task File Table (TFT) is a job-related table which establishes the connection between a physical file and a program. It consists of entries containing information on the file attributes and volumes. This information is taken into account during file processing. A TFT entry is accessed via its file link name.

If a link name for a file is specified by means of the LINK operand in the FILE command, the operating system creates an entry with this name in the TFT of the job and transfers to this entry all the values specified in the FILE command, including any NULL operands (see below). A value specified in the TFT entry is given precedence over a value in the FCB. If neither the TFT entry nor the FCB contains a value, or if the FILE command was specified with a NULL operand, then the value from the catalog or, if a new file is created, the appropriate default value is used.

When the file is opened, these values are transferred to the file control block (FCB).

When the file is closed, the catalog entry is updated with the values from the file control block.

### Tape Set Table (TST)

The TST is a job-related table which, in conjunction with the TFT (see above) establishes the connection between volumes and files. It consists of entries containing a list of volumes. These entries can be accessed via their names.

TST entries are created with `FILE ..., TSET=name` and released with `RELEASE linkname`.

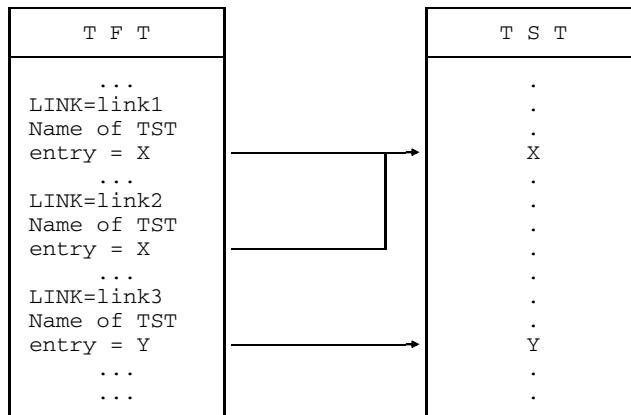


Fig. FILE-1 Relationship between TFT and TST

### Pool link name / ISAM pools

With NK-ISAM, ISAM files are processed in ISAM pools. The connection between the user ISAM pool and the file is established via the pool link name, which is specified by means of the POOLLNK operand.

### File attributes / volumes

The FILE command allows file attributes such as record length, block length, etc. to be defined. In addition, devices and volumes can be requested, and storage space assigned or released.

The operand description refers to special features and interaction between the various operands.

### NULL operands

If the file/file generation specified in the FILE command already exists, the operands FCBTYP, RECFORM, RECSIZE, BLKSIZE, KEYLEN, KEYPOS, LOGLEN, VALLEN and VALPROP may be specified as "NULL operands" together with a file link name. A NULL operand is an operand for which no value is specified in the command call. For example:

```
/FILE ..., LINK=name, FCBTYP=, RECFORM=, ...
```

The corresponding fields of the TFT entry then contain the character string "NULL" as their value (output after the RDTFT command). When the file is opened, the information for these file attributes is transferred from the catalog entry to the file control block (FCB).

### **Processing of ISAM files with separate index and data sections**

ISAM files can be created with separate index and data sections on different private disks. Both sections of the file may be independently assigned devices, volumes, and storage space - using DEVICE, VOLUME, and SPACE for the index section, and DDEVICE, DVOLUME, and DSPACE for the data section. The allocation of devices, volumes, and storage space is carried out according to the same principle; however, it must be noted that the entries for SPACE always refer to VOLUME and DEVICE, while those for DSPACE always refer to DVOLUME and DDEVICE. Only the disk device types valid under DEVICE are permitted as device types for DDEVICE. There is no default value for DSPACE when a file is created.

If the FILE command refers to a file that does not yet occupy storage space, then DDEVICE, DVOLUME and DSPACE must always be specified together. In the case of files that already occupy storage space, DSPACE may also be specified without DDEVICE and DVOLUME. Storage space can only be released for the entire file (with SPACE); it cannot be released separately for the index and data section. Once an ISAM file has been created with a separate index and data section, the index and data blocks can no longer be copied to the same volume. The option to split an ISAM file in this way is available on private volumes only, not on public volumes.

With NK-ISAM, it is not necessary to separate the index section and the data section. A secondary allocation only affects the data section.

### **Processing tape files**

When a tape file is created, unlike a disk file, it is cataloged by default as shareable (SHARE=YES). However, you can use the CATALOG command to change the file attributes before creating it (OPEN OUTPUT/OUTIN). The modified file attributes are entered in the file labels before the file is opened.

If successive file generations within a group belong to the same MF/MV set, DISP=REUSE must on no account be specified in the CATALOG command since this may cause file generations to be destroyed.

Fig. FILE-2 Functions of the FILE command

## Function overview of the FILE command

### 1. Naming and cataloging files, defining link names

Operand	Operand value	Function
pathname		<ul style="list-style-type: none"> <li>- Create a catalog entry</li> <li>- Allocate storage space for disk files (primary allocation)</li> <li>- Name the file/catalog entry to which subsequent operands will refer</li> </ul>
*DUMMY		Define a dummy file (primarily for test purposes)
LINK	name	Define a file link name, for which a TFT entry is created
POOLLNK	name	For NK-ISAM files: define a pool link name for the user ISAM pool
STATE	FOREIGN	Import a non-cataloged file from private volumes

### 2. Defining file attributes

Operand	Operand value	Function
FCBTYPE	ISAM/PAM/SAM/ BTAM	Access method for the file
BLKCTRL	PAMKEY	File format "PAMKEY": block control information is entered in a key field separate from the PAM block
	DATA	The block control field is located at the start of the PAM block
	NO	No block control information (converted to BLKCTRL=DATA for ISAM and SAM)

Operand	Operand value	Function
RECFORM	V	File with "variable" length records; the maximum record length is defined with RECSIZE
	F	File with "fixed" length records; the record length is specified with RECSIZE
	U	Record with "undefined" length records; the length of each record is entered in a register; each data block contains one, and only one, record
	N/M/A	Specifies whether printer control characters are to be taken into account at specific positions
RECSIZE	length	Record length for RECFORM=F/V
	r	Register which contains the length of the current record for RECFORM=U
BLKSIZE	(STD,number)	Block size as a multiple of the standard PAM block
	length	Tape files: block size for non-standard blocks
KEYPOS	number	ISAM files: start position of the primary key in the data record
KEYLEN	number	ISAM files: length of the primary key
LOGLEN	number	ISAM files: length of the logical flag in the ISAM index
VALLEN	number	ISAM files: length of the value flag in the ISAM index
CODE	EBCDIC/ISO7/ OWN	Tape files: code with which the file was created

### 3. Requesting devices and volumes

Operand	Operand value	Function
DEVICE	device	Define device type
	WORK	Request work tape
VOLUME	(PRIVATE,n)	Request private volumes
	(vsn,...)	Define volume list
MOUNT	(number,...)	Mount request for private volumes
DDEVICE	device	K-ISAM files only: define volume type for data section
DVOLUME	(vsn,...)	K-ISAM files only: define private disk(s) for data section
	(PRIVATE,n)	Request, at the console, for n volumes to be mounted
FSEQ		For tape files, when several files are on a single tape
	UNK	The file position within a file set is unknown
	NEW	The file is added to the file set
VSEQ	number	Position of the file in the file set
		For tape files that extend over multiple tapes
	number	designates the n-th tape plus all following tapes within the volume list
	L=(number,...)	Specifies the desired file section, i.e. the n-th tape of the volume list
TSET	(name,vsn)	Tape files: defines a tape set for extending files or file sets
TVSN	(vsn,...)	Tape files: specifies which tapes from the tape set are required for current processing



#### 4. Open mode and processing attributes

Operand	Operand value	Function
OPEN	INPUT/OUTPUT/ EXTEND/INOUT/ OUTIN/UPDATE/ SINOUT/REVERSE	Specifies the OPEN mode for the file
SHARUPD	NO/YES	PAM or ISAM files: shared-update processing permitted or not permitted
CLOSMG	NO/YES	SAM files: output of a message after completion of CLOSE processing
WRCHK	NO/YES	Disk files: read-after-write check
OVERLAP	NO/YES	ISAM files: overlapped processing with regard to the I/O buffer
PAD	number	ISAM files: free space in data blocks during sequential file creation
DUPEKY	NO/YES	ISAM files: Specifies whether or not the same key value may occur in a number of records
VALPROP		K-ISAM files: controls the evaluation of the value flags in a block for the key entry of the next higher level
WROUT	NO/YES	ISAM files: write updated blocks immediately
LABEL	(STD,number)	Tape files: file with standard labels (as per DIN 66029)
	NO	Tape file without labels
	NSTD	Tape file with non-standard labels
TPMARK	YES/NO	Tape files: write tape marks
DESTOC	NO/YES	Tape files: "destruction" of remaining data on tape

Operand	Operand value	Function
TRANS	YES/NO	Tape files: convert non-EBCDIC tape files
BUFOFF	L/length	Tape files: length of buffer offset within the data block
TAPEWR	DEVICE-BUFFER	Files on tape cartridges: buffered processing
	IMMEDIATE	Unbuffered processing
CHAINIO	number	Tape files: chained input/output
BLIM	number	Tape files: automatic tape changing and checkpointing for n blocks
WRCPT		Tape files: write checkpoint
SECLEV	HIGH/LOW	Tape files: ignore file protection attributes under specific conditions (TPIGNORE attribute in the JOIN entry)
RETPD	days	File protection for the specified period
BYPASS	LP/(LP, [;-]n)	Tape files: bypass label checking for input files (authorized users)

## 5. Space management for disk files

Operand	Operand value	Function
SPACE	primary	Primary allocation: space allocation takes immediate effect primary > 0: storage space is reserved primary < 0: storage space is released primary = 0: no storage space is reserved; only a catalog entry is created
	secondary	Secondary allocation: space allocation is extended as required
	(page, number, ABS)	Files on private disks: absolute allocation; "page" is the position of the file extent, "number" the number of PAM pages reserved
DSPACE	same as SPACE	ISAM files: space management for the data section of files on private disk, if the index and data sections are separated (see SPACE; space allocation only refers to the data section of the ISAM file - for space allocation for the index section, the SPACE operand must be used)

## Format and operand description

Operation	Operands
FILE	<p>[ {pathname} ] [ , LINK=name ] [ , POOLLNK=name ] [ , STATE=FOREIGN ]</p> <p>[ *DUMMY ]</p> <p>[ , DEVICE = { device } ] [ , VOL[UME] = { PRIVATE (PRIVATE, n) vsn (vsn, ...) } ] [ , MOUNT = { number (number, ...) } ]</p> <p>[ , SPACE = { primary (primary [ , secondary] ) } ]</p> <p>[ , OPEN = { INPUT EXTEND INOUT OUTPUT REVERSE SINOUT } ] [ , FCBTYP = { ISAM BTAM PAM SAM } ] [ , BLKCTRL = { PAMKEY DATA NO } ]</p> <p>[ , RETPD=days ] [ , SHARUPD = { NO YES } ] [ , CLOSMSG = { NO YES } ] [ , WRCHK = { NO YES } ]</p> <p>[ RECFORM = { { V F U } ( { V F U } [ , { N M A } ] ) } ] [ , RECSIZE = { length reg } ] [ , BLKSIZE = { STD (STD, n) length } ]</p> <p>[ , KEYPOS=number ] [ , KEYLEN=length ] [ , DUPEKY = { NO YES } ]</p> <p>[ , LOGLEN=length ] [ , VALLEN=length ] [ , VALPROP = { MIN MAX } ]</p> <p>[ , OVERLAP = { NO YES } ] [ , PAD=number ] [ , WROUT = { NO YES } ]</p>

Operation	Operands
FILE (cont.)	$[ , DDEVICE=device ] [ , DVOLUME= \left. \begin{array}{l} \text{PRIVATE} \\ \text{(PRIVATE, n)} \\ \text{(vs n, ...)} \\ \text{(vs n, ...)} \end{array} \right\} ] [ , DSPACE= \left. \begin{array}{l} \text{primary} \\ \text{(primary [ , secondary] )} \\ \text{(page, number, ABS)} \end{array} \right\} ]$ $[ , LABEL= \left. \begin{array}{l} \text{(STD, 3)} \\ \text{STD} \\ \text{(STD, number)} \\ \text{NO} \\ \text{NSTD} \end{array} \right\} ] [ , TPMARK= \left. \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} ] [ , DESTOC= \left. \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} ]$ $[ , CODE= \left. \begin{array}{l} \text{EBCDIC} \\ \text{ISO7} \\ \text{OWN} \end{array} \right\} ] [ , TRANS= \left. \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} ]$ $[ , FSEQ= \left. \begin{array}{l} \text{UNK} \\ \text{NEW} \\ \text{number} \end{array} \right\} ] [ , VSEQ= \left. \begin{array}{l} \text{number} \\ \text{(L= (number, ...))} \end{array} \right\} ]$ $[ , TSET= \left. \begin{array}{l} \text{name} \\ \text{(name, vsn)} \end{array} \right\} ] [ , TVSN= \left. \begin{array}{l} \text{vs n} \\ \text{(vs n, ...)} \end{array} \right\} ]$ $[ , BUFOFF= \left. \begin{array}{l} \text{L} \\ \text{length} \end{array} \right\} ] [ , TAPEWR= \left. \begin{array}{l} \text{DEVICE-BUFFER} \\ \text{IMMEDIATE} \end{array} \right\} ] [ , CHAINIO=number ]$ $[ , BLIM=number ] [ , \left. \begin{array}{l} \text{CHKPT} \\ \text{CP} \end{array} \right\} = \left( \left. \begin{array}{l} \text{NO} \\ \text{ANY} \\ \text{BLIM} \\ \text{FEOV} \end{array} \right\} , \left. \begin{array}{l} \text{ACTIVE} \\ \text{DUMMY} \end{array} \right\} \right) ]$ $[ , SECLEV= \left. \begin{array}{l} \text{HIGH} \\ \text{LOW} \\ \left( \left. \begin{array}{l} \text{HIGH} \\ \text{LOW} \end{array} \right\} , \text{OPR} \right) \end{array} \right\} ] [ , BYPASS= \left. \begin{array}{l} \text{LP} \\ \text{(LP, } \left. \begin{array}{l} \text{n} \\ \text{+n} \\ \text{-n} \end{array} \right\} \text{)} \end{array} \right\} ]$



The default values for the operands are interpreted and entered in the TFT entry only in conjunction with the LINK operand. They are not valid unless neither the corresponding field in the FCB nor the one in the TFT has been supplied with a value.

pathname	stands for [:catid:][\$userid.]filename  Designates the file or file generation to which this FILE command refers; "pathname" must not be a file generation group. If "pathname" is not yet cataloged, a catalog entry is created and the space specified for the primary allocation (see SPACE, below) is allocated.
catid	Catalog ID of the pubset on which the file is stored. Default value: the catalog ID associated with the user ID (taken from the JOIN entry).
userid	User ID of the file. Default value: user ID specified in the LOGON command.  If the specified user ID differs from the one in the LOGON command, "pathname" must already be cataloged as shareable. Otherwise, the FILE command will be rejected.
filename	Fully-qualified name of a file or file generation.
*DUMMY	Describes a dummy file. If the LINK operand is specified, a TFT entry with a volume list is created. If the TSET operand is also specified, a TSET entry is created in addition to the TFT entry. All other operands are simply checked for formal correctness but otherwise ignored, which means that neither devices nor volumes are requested, nor is storage space allocated, and no catalog entry is created.  Dummy files can, for example, be used for simulation of I/O operations during program testing or, for RESTART processing, replace files which are not actually needed for processing but are specified in the program.  Dummy file as an input file: when the program attempts to read from the file, EOF processing is initiated. Dummy file as an output file: data is transferred to the I/O areas of the program, but output to a volume is suppressed.

- BLIM=number** Only for creating tape files with standard labels which are to be processed with the SAM access method and extend over several tapes. This means that "pathname" must not yet be cataloged, and the following operands must be specified in the FILE command: FCBTYPE=SAM, OPEN=OUTPUT, LABEL=(STD,n), VOLUME=(vsn,...).
- "number" specifies how many data blocks may be written to one tape, where  $1 \leq \text{number} \leq 999999$ . When this limit is reached, a tape swap is initiated (EOV processing). If requested with the CHKPT operand, a checkpoint is written to the end of the tape before EOV processing is started. If the end of the tape is reached before the specified number of blocks has been written, the user receives an error message in the FCB.
- BLKCTRL** Defines the file format (see also the manual "DMS Introductory Guide and Command Interface" [8]).
- Default values:
- |                        |                 |
|------------------------|-----------------|
| Volume with PAMKEY:    | BLKCTRL=PAMKEY; |
| Volume without PAMKEY: |                 |
| SAM/ISAM files:        | BLKCTRL=DATA;   |
| UPAM files:            | BLKCTRL=NO.     |
- BLKCTRL can also be specified as a NULL operand, in which case the BLKCTRL information from the catalog entry of the file is taken over when the file is opened.

The table below shows the interaction of the BLKCTRL operand and the operands for ISAM file processing.

Operand	BLKCTRL	
	= PAMKEY	= DATA / NO
POOLLNK	No ISAM pools for K-ISAM files	Connects the file to a user ISAM pool (otherwise: standard ISAM pool)
PAD	Minimum value for free space in data block	Maximum space to be left free in each data block
DUPEKY		Records with duplicate keys have a time stamp added internally
WROUT	Default value depends on SHARUPD	Default value depends on SHARUPD and ISAM pool
SHARUPD	Block locks	Record or range locks
VALPROP	Value flag is evaluated for index entry	Value flag is ignored
DDEVICE DVOLUME DSPACE	Index and data stored and managed separately on private disks	Separation of index and data sections not supported

- =PAMKEY**      K file; the file format is "key-specific", i.e. the system stores control information in the PAMKEY, provided the volume allows the PAMKEY to be recorded.
- =DATA**        NK file; the file format is "no-key", i.e. the system does not store control information in the PAMKEY but at the beginning of every logical block (in the case of ISAM files at the beginning of every 2-K block).
- =NO**            NK file; the file format is "no-key", i.e. the system does not store control information in the PAMKEY. This file format is for UPAM files only. SAM and ISAM files are processed as described for BLKCTRL=DATA.



**BLKSIZE** Specifies the blocking factor (for K files) or the length of a logical block/data block (for NK files), i.e. the length of the data transfer unit between the I/O devices.

The BLKSIZE specification is always interpreted and shown as BLKTYPE and BLKSIZE in the output of the FSTAT command.

For disk files, there is interaction between this operand and the SPACE and RECSIZE operands; for tape files, between this operand and the LABEL operand.

In the case of disk files/tape files with standard blocks, data blocks may consist of several PAM pages. The system automatically links together the PAM pages belonging to one transfer unit.

For tape files with non-standard blocks, the block format is not the same as that used by PPAM; a data block is defined as the number of bytes which are read or written in one read or write operation.

=STD

The file consists of standard PAM pages; default value. The data block and the PAM page are identical. The usable length of the standard block depends on the BLKCTRL specification (or on the disk type). Standard block lengths are:

BLKCTRL	PAM page and usable length
= PAMKEY	2048-byte PAM page + 16-byte PAM key
= DATA	2048-byte PAM page - 16-byte block control field → 2032 bytes usable block length
= NO	2048-byte PAM page = usable block length

=(STD,n)

For K files: "n" is the blocking factor ( $1 \leq n \leq 16$ ): each data block consists of n PAM pages, which means that the maximum block size is 16 PAM pages = 32768 bytes.

For NK files: the length of the logical block is  $n * 2048$ .

=length

Only for tape files: specifies the block length in bytes and, at the same time, specifies that the file consists of non-standard blocks, i.e. other than PAM pages.

When specifying "length", the user must consider, on the one hand, the settings of BUFOFF and RECFORM and, on the other hand, the settings FCBTYP and CHAINIO.

## Relationship between block length and record format:

Operand RECFORM	Effects
RECFORM=F	"length" specifies the block size including the length of any buffer offset (see the BUFOFF operand). All blocks are the same size.
RECFORM=V/U	"length" specifies the maximum block size including the length of any buffer offset (see the BUFOFF operand). The block size, just like the record length, is variable.  If RECFORM=V is used together with CODE=EBCDIC or LABEL=(STD,n) (n>1), "length" must be less than 100000

## Relationship between access method and block length:

Operand FCBTYPE	Permissible values for "length"
SAM/BTAM	$1 \leq n \leq 32768$
PAM	_____

## BUFOFF

Only for tape files with BLKCTRL=DATA or SAM tape files without standard blocking. Defines the buffer offset, i.e. the length of a field which is inserted at the beginning of each data block.

Default value for tape files with BLKCTRL=DATA:

for FCBTYPE=SAM: BUFOFF=16;

for FCBTYPE≠SAM:

BUFOFF=12.

Default value for SAM tape files without standard blocking:

for RECFORM=V: BUFOFF=4;

for RECFORM=F: BUFOFF=0.

Buffer offset is permissible only together with RECFORM=V/F or LABEL=(STD,2)/(STD,3). If neither of these is specified, the specification of BUFOFF will lead to an error.

- =L            The BUFOFF value is taken from the HDR2 label of the file. If there is no HDR2 label, or if the "buffer offset" field contains blanks (X'4040'), the default value is used.
- =length       Specifies the length of the buffer offset.
- For SAM files with RECFORM=V:  $0 \leq \text{length} \leq 4$ ; if BUFOFF=4, the initial field contains the length of the current block; in the case of files with BLKCTRL=DATA, it contains the block control field.
- BUFOFF can also be specified as a NULL operand, in which case the BUFOFF information from the catalog entry of the file is taken over when the file is opened.
- BYPASS       Only for input files on tape. If permitted by the JOIN entry for your user ID, you can bypass the label checking routines and specify how the tape is to be positioned. DMS checks that the correct tape is mounted and activates any user routines for label handling in the normal manner. The positioning specification is evaluated only if no OPENV exit is defined.
- In addition to label checking, code checking is also bypassed. If CODE=OWN is specified, the user must provide appropriate code tables of his own.
- BYPASS permits the processing of tapes created under other operating systems (such as BS1000) or of tapes whose structure and label formats are not known to the system. The BYPASS specification is valid only during file processing; it is not included in the catalog entry for the file.
- If specified together with BYPASS, the FSEQ operand is not evaluated.
- =LP            Label handling is not executed. The header labels are neither checked nor read. The tape position is not changed.
- =(LP,n)        $0 \leq n \leq 32767$ . Labels are not handled. When the file is opened, the tape is positioned to the nth tape mark from the beginning of the tape.
- (LP,0): the tape is positioned to the beginning of the tape.
- =(LP,+n)       $0 \leq n \leq 127$ . Labels are not handled. When the file is opened, the tape is advanced by n tape marks from its current position.
- (LP,+0): the tape is not repositioned.

=(LP,-n)  $0 \leq n \leq 127$ . Labels are not handled. When the file is opened, the tape is repositioned by n tape marks from its current position.

(LP,-0): the tape is not repositioned.

CHAINIO=number

Only for BTAM files with chained I/O.  $1 \leq \text{number} \leq 16$ ; "number" is the chaining factor which defines the length of the transport/transfer unit for input and output. "number" is a number of physical blocks, which means that the length of the transport unit is "number" \* BLKSIZE.

Although "number" can be overwritten by specifications in the program when processing BTAM files, CHAINIO=number must still be specified if chained I/O is to be used.

{CHKPT}  
{CP }

For tape files: controls whether and when a checkpoint is to be written automatically at the end of the tape or how file processing is to continue after a restart (RESTART command).

Default value: CHKPT=(NO,ACTIVE)

=(NO,) No checkpoints are written automatically.

=(BLIM,) When the block limit specified by the BLIM operand is reached, a checkpoint is written automatically; the BLIM operand must be specified.

=(FEOV,) A checkpoint is written automatically each time the FEOV macro is called in an Assembler program.

=(ANY,) A checkpoint is written automatically when the BLIM limit is reached or when the FEOV macro is called in an Assembler program. The BLIM operand must be specified.

=(NO,DUMMY) "pathname" is treated like a dummy file in the case of a restart using the RESTART command.

=(...,ACTIVE) The file "pathname" is processed further in the case of a restart using the RESTART command.

CLOSMMSG For files to be processed sequentially (SAM): you can specify that a message is to be issued to SYSOUT on completion of CLOSE processing.

Default value: for disk files: CLOSMMSG = NO

for tape files: CLOSMSG = YES

=NO	The close message is suppressed.
=YES	The close message is issued.
CODE	<p>For SAM and BTAM tape files: specifies whether code translation tables are to be used during input/output.</p> <p>For CODE=EBCDIC and CODE=ISO7 the German and international character sets are encoded in the same manner.</p> <p>For CODE=ISO7/OWN and FCBTYP=SAM, the following should be noted:</p> <ul style="list-style-type: none"> <li>– the block size must be specified with BLKSIZE=length, so that no PAM keys are written;</li> <li>– for outputs in locate mode (see the manual "DMS Introductory Guide and Command Interface" [8]) with variable-length records (RECFORM=V), the contents of the record length field change.</li> </ul> <p>CODE can also be specified as a NULL operand, in which case the CODE information from the catalog entry of the file will be taken over when the file is opened.</p>
=EBCDIC	No code conversion during processing is necessary; default value.
=ISO7	The tape file is written in ISO 7-bit code, which means that EBCDIC code is converted to ISO 7-bit code for output and vice versa for input.
=OWN	Conversion is carried out with code tables provided by the user. The addresses of these tables must be in the file control block. At the same time, label processing must either be deactivated with LABEL=NO or it must be carried out in the user program with LABEL=NSTD.
DDEVICE=device	For K-ISAM files with index and data sections stored on different private disks: DDEVICE designates the disk type for the data section (that for the index section is specified with DEVICE); permissible entries for "device" can be found in the device table in the appendix. DDEVICE must be specified if no storage space has yet been reserved for the file.

DESTOC	<p>For tape processing: defines whether any remaining data on the tape is to be erased by overwriting on completion of EOF/EOV processing.</p> <p>DESTOC is only effective if a TFT entry is also created for "pathname" by means of the LINK operand.</p> <p>The DESTOC specification has the same function as the DESTROY operand in the CATALOG command. However, DESTOC overrides the entry in the DESTROY field of the catalog entry. If DESTOC is omitted in the FILE command, the specification from the catalog entry is used.</p>
=NO	Data on the remainder of the tape is not deleted.
=YES	After the EOF labels have been written, the data on the remainder of the tape is erased.
DEVICE	<p>For files on private volumes: defines the device type for the volume (disk/tape) or, in the case of tape processing, requests a work tape.</p> <p>Default value:      If neither DEVICE nor VOLUME is specified, and a new file is to be created, the file will be created on public disk; otherwise, in the case of an existing file, the volume type in the catalog entry is used.</p>
=device	<p>Specifies the type of volume; permissible entries for "device" in the case of tape or disk devices can be found in the device list in the appendix.</p> <p>If, when a file is created, a tape device is specified for DEVICE, but no VOLUME is indicated, a free tape with standard labels will be requested and assigned to the operator when the file is opened. A free tape is a tape on which nothing has yet been written or any tape for which the retention period of the first file on the tape has expired, so that write access is permitted.</p>
=WORK	<p>Only for tape processing: requests a work tape. Such tapes are not assigned to owners (their VOL1 labels contain only blanks, i.e. X'40') and are automatically released at the end of the job. It is therefore not possible to protect and archive files on a work tape.</p> <p>If DEVICE=WORK is specified, any entries for the VOLUME operand are ignored. The operands STATE=FOREIGN and TSET must not be specified together with DEVICE=WORK.</p>

DEVICE=WORK should not be specified for multivolume files, since any available work tape is automatically assigned.

Magnetic tape cartridges cannot be requested as work tapes.

**DSPACE** Defines the space allocations for the data section of a K-ISAM file (for K-ISAM files whose index and data sections are stored on different private disks). The conventions for primary, secondary and absolute allocation are the the same as those for the SPACE operand. However, the entries refer to the volume(s) specified in the DVOLUME operand (see also the DDEVICE and DVOLUME operands).

=primary Primary allocation; effective immediately.  
 $0 \leq \text{primary} \leq 50331645$

=(primary,secondary) Primary allocation, which is effective immediately, and secondary allocation, which is transferred to the catalog entry.  
 $0 \leq \text{secondary} \leq 32767$

=(page,number,ABS)  
 page: number of the PAM page at which the allocation begins;  
 number: number of PAM pages to be allocated;  
 ABS: absolute allocation.

**DUPEKY** For ISAM files; specifies whether duplicate ISAM keys may be used. DUPEKY is only significant if the file is to be processed with the PUT or STORE macro (See "DMS Assembler Interface" manual [9] and "Executive Macros" manual [5]).

=NO Duplicate ISAM keys are not permitted; default value.

=YES Duplicate ISAM keys are permitted. Records that have the same ISAM key do not overwrite each other, but are written sequentially in the order in which they are created.



In NK-ISAM, an 8-byte time stamp is appended internally to records with duplicate primary keys. This must be taken into account when the record length is defined. "Secondary" keys may also be defined - in which case no duplicate primary keys may be used.

DVOLUME	For K-ISAM files with index and data sections stored on different private disks; specifies the volume serial number ("vsn") of the volume on which the data section of the ISAM file is to be stored. The operand VOLUME must be specified for the index section.
=vsn	The specified private disk is requested for the data section of the ISAM file.
=(vsn,...)	Several private disks, whose VSNs are listed, are requested. A maximum of 255 VSNs may be specified.
=PRIVATE	Requests a private volume at the console.
=(PRIVATE,n)	$1 \leq n \leq 9$ ; requests "n" private volumes at the console.
FCBTYPE	Specifies the access method to be used for file processing.  FCBTYPE can also be specified as a NULL operand, in which case the FCBTYPE information from the catalog entry of the file will be transferred when the file is opened.
= <u>ISAM</u>	"pathname" is an ISAM file; default value. Depending on the value of the BLKCTRL operand, it is processed as an NK-ISAM file (BLKCTRL=DATA) or as a K-ISAM file (BLKCTRL=PAMKEY).  ISAM-specific operands: DUPEKY, KEYLEN, KEYPOS, LOGLEN, VALLEN, WROUT and DDEVICE, DSPACE, DVOLUME and VALPROP.
=BTAM	"pathname" is a tape file which is to be processed using access method BTAM. BTAM processing is not supported by all programming languages.  BTAM-specific operands: CHAINIO, OPEN=SINOUT
=PAM	"pathname" is a PAM file that is processed using access method UPAM. PAM files may be stored on tape or disk.
=SAM	"pathname" is a SAM file on disk or tape. SAM files are generally processed sequentially using either of the access methods SAM or UPAM.  SAM-specific operands: BUFOFF, CLOSMSG, OPEN=UPDATE



FSEQ	<p>For tape files which belong to a file set: specifies the (sequential) number of a file within the file set. If, for example, several files with the same name are stored on one tape, access to a specific file is controlled by means of FSEQ. This also applies to MF/MV sets.</p> <p>Default value: FSEQ = 1</p> <p>FSEQ can also be specified as a NULL operand, in which case the FSEQ information from the catalog entry of the file will be taken over when the file is opened.</p>
=UNK	<p>Permissible only for files with standard labels: The start position of the file is unknown.</p>
=NEW	<p>Permissible only for files which are not yet cataloged: The file "pathname" is to be added to a file set.</p> <p>The file pointer is positioned to the end of the file set, and the new file is written behind the currently last file of the file set. The "file sequence number" is incremented by 1.</p>
=number	<p>Specifies the file sequence number of "pathname" within the file set, where <math>0 \leq \text{number} \leq 9999</math>.</p> <p>FSEQ=0, just like FSEQ=1, denotes the first file of the file set.</p> <p>If "pathname" is already cataloged, "number" must be the same as the file sequence number in the catalog entry. If a new file is to be created, it is added to the end of the file set, which means that its file sequence number must be 1 higher than that of the "old" last file in the file set. "number" is transferred to the catalog entry (even in conjunction with STATE=FOREIGN).</p>
KEYLEN=length	<p>For ISAM files; specifies the length of the ISAM primary key in bytes, where <math>1 \leq \text{length} \leq 255</math>.</p> <p>Default value: KEYLEN = 8</p> <p>KEYLEN can also be specified as a NULL operand, in which case the KEYLEN information from the catalog entry of the file will be taken over when the file is opened.</p>

**KEYPOS=number** For ISAM files; specifies the position of the ISAM primary key in the record. In variable-length records, 4 bytes for the record length and control field must be taken into account. The ISAM primary key may be anywhere in the record, but must be in the same position in each record of one and the same file.

Default value:

for files with RECFORM=V: KEYPOS=5

for files with RECFORM=F: KEYPOS=1

KEYPOS can also be specified as a NULL operand, in which case the KEYPOS information from the catalog entry of the file will be taken over when the file is opened.

**LABEL**

Only for tape files; specifies the label type for files on tape or tape cartridge; the SECLEV operand determines how the labels are processed.

Default value: LABEL=(STD,3)

For existing tape files, the label standard version in the VOL1 label always applies. The LABEL operand is evaluated for output files (OPEN OUTIN/OUTPUT). If the tape already contains files or file sections, the label standard version in the VOL1 label is set or updated as specified in the LABEL operand.

=STD

File and volume receive/have standard labels in accordance with DIN 66029, interchange level 1.

=(STD,number)

File and volume receive/have standard labels in accordance with the DIN 66029 interchange level designated by "number", where  $0 \leq \text{number} \leq 3$ ; interchange level 4 is in preparation.

The following table illustrates the effects of the LABEL operand:

	(STD, 0)	(STD, 1)	(STD, 2)	STD, 3)
DIN 66029 interchange level/ date	-	1 8/1972	2 6/1976	3 3/1978
Label standard version in VOL1 label	_ (blank)	1	2	3
CODE=ISO-7/OWN	not permitted	STD blocks converted to non-standard blocks  RECFORM=V: conversion to D-format  RECSIZE > 9999 or BLKSIZE > 9999 OPEN error	STD blocks converted to non-standard blocks	STD blocks converted to non-standard blocks
CODE=EBCDIC			STD blocks converted to non-standard blocks	STD blocks converted to non-standard blocks
Access method			SAM only	SAM only
RECFORM=U				Invalid for output files; converted to (STD,2)

(STD,1) is used for:

- RECFORM=V and CODE=EBCDIC
- BLKSIZE=STD
- FCBTYP= PAM or FCBTYP=BTAM

For (STD,0), CODE=EBCDIC must be used.

If the label standard version in the VOL1 label is less than (STD,number), "number" assumes the value of the label standard version.

=NO

File labels are neither read nor written (no file label processing). If the tape has standard labels, the system processes the volume labels and checks the access rights.

=NSTD            The tape file already has or is to receive non-standard labels and file label processing is performed in the user program. If the volume has standard labels, the system processes these and checks the access authorization.

LOGLEN=length    For K-ISAM files only; specifies the length (in bytes) of the logical flag in the ISAM index. The maximum length is determined by the length of the primary key and the length of any existing value flag (see VALLEN, below). Since the entire ISAM index must not be longer than 255 bytes, the following rule applies:

$$\text{length} \leq 255 - \text{KEYLEN} - \text{VALLEN}$$

Default value:    LOGLEN=0, i.e. there is no logical flag in the ISAM index.

In the ISAM index, the primary key may be followed by a logical flag in which selection criteria are defined bit-by-bit and encoded in binary code. All logical flags of a block are evaluated, and the result is placed in the next-higher index entry. The LOGLEN specification is ignored by NK-ISAM.

LOGLEN can also be specified as a NULL operand, in which case the LOGLEN information from the catalog entry of the file will be transferred when the file is opened.

LINK=name        A link can be established dynamically in the TFT entry between a file and a processing program by means of the file link name. Here, a TFT entry is to be created for the specified file link name ("name"). The other operands are then evaluated (incl. NULL operands) and their values are placed in this TFT entry (exception: SPACE operand).

If an entry of the same name already exists in the TFT, this entry is released and then recreated. The old device reservations are cancelled except for tape devices, which remain reserved for the job and may need to be released with the SECURE command. If the old TFT entry was locked with the HOLD command, the new entry will be locked as well.

If the LINK operand is not specified, no TFT entry is created, as a result of which some of the specified operands cannot be interpreted (see "Processing of catalog entries" on page 209).

"name" may be from 1-8 characters; permissible character set: letters, digits, special characters in accordance with the character set available for file names. The name must contain at least one letter or one special character; "." and "-" must not be specified next to each other; a period must not be used at the beginning of the name.

## MOUNT

Determines whether or not the operator at the console should receive a request for the private volume(s) specified.

If "pathname" refers to a tape file, the operator at the console receives a request for the private volumes. The volume serial numbers are obtained from the volume list in the catalog entry and from the specifications in the VOLUME operand. This request is suppressed when MOUNT=0 is specified (see below).

If "pathname" refers to a disk file, the MOUNT operand is ignored. Exception: MOUNT=0 (see below).

The default value is MOUNT=1, which causes the first volume in the volume list to be requested.

### *Interaction between MOUNT and other operands:*

- If DEVICE=WORK is specified at the same time, the MOUNT operand is ignored.
- MOUNT values must not be less than those of VSEQ (exception: MOUNT=0);  
if VSEQ=n, the MOUNT list must begin with "n":  
(MOUNT=(n[,n+1][,n+2][,...]));  
if VSEQ=(L=(n1, n2,...)), the VSEQ and MOUNT lists must match  
(MOUNT=(n1, n2,...)); the MOUNT list may contain fewer elements.  
If VSEQ is omitted, the MOUNT list must begin with 1 and ascend without gaps.
- If TSET and VOLUME are also specified for non-cataloged tape files, the operating system applies the MOUNT operand on the volume list in the TST entry.
- If VOLUME is also specified for cataloged tape files, MOUNT refers to the volume list in the catalog entry and to the logically following list in the VOLUME operand; for non-cataloged tape files, MOUNT refers only to the list in the VOLUME operand.

- =0 If "pathname" refers to a disk file, the volume is not requested until the file is to be opened. "pathname" must exist and neither DEVICE, VOLUME nor SPACE (or DDEVICE, DVOLUME, DSPACE) may be specified.
- =number  $1 \leq \text{number} \leq 255$  (number > 1 together with VSEQ only). This specification refers to the sequence of the volume serial numbers in the volume list. MOUNT=n causes the nth tape in the list to be requested and a tape device to be made available.
- =(number,...)  $1 \leq \text{number} \leq 255$  (number > 1 together with VSEQ only). A number of volumes to be mounted can be requested in a list. The numbers in the list must be in ascending order. One tape device is then reserved for each volume specified in the list.

OPEN Specifies the OPEN mode for the file. This setting may be overwritten by the call to the OPEN macro in the program when the file is opened.

The following table shows which OPEN modes are permissible for the various access methods.

OPEN mode	ISAM	BTAM	SAM	UPAM
INPUT	x	x	x	x
EXTEND	x	x	x	-
INOUT	x	x	-	x
OUTIN	x	x	-	x
OUTPUT	x	x	x	-
REVERSE	-	x	x	-
SINOUT	-	x	-	-
UPDATE	-	-	x	-

x  $\triangleq$  OPEN mode is permitted

-  $\triangleq$  OPEN mode is not permitted

The various OPEN modes are described in detail in the descriptions of the access methods.

=INPUT "pathname" is an input file, i.e. it must exist; default value.

=EXTEND An existing file is extended, i.e. further data blocks are added to the end of the file or the file is overwritten from a certain position onwards; only sequential write operations are permitted. For tape files, the value in the LABEL operands determines whether or not labels are to be generated.

=INOUT	An existing file is opened for non-sequential processing; write and read operations are permitted. For tape files, the tape is positioned to the beginning of the tape after OPEN processing; no labels are written.
=OUTIN	A file is created or, if it already exists, overwritten from the beginning. Both read and write operations are permitted (non-sequential). Labels are written for tape files.
=OUTPUT	A file is created or, if it already exists, overwritten from the beginning. Labels are created for tape files.
=REVERSE	The file "pathname" must already exist and is used as an input file for sequential reading from end-of-file to beginning-of-file. The file must not extend over several volumes. For tape files, the tape is positioned to the end of the file after OPEN processing.
=SINOUT	Only for tape files processed with the BTAM access method. The file must exist and the tape must not be positioned to the beginning of tape. Data blocks can be read or written; labels are not processed. In contrast to INOUT, the tape is not positioned.
=UPDATE	Only for SAM disk files; the file is to be processed in locate mode.
OVERLAP	<p>Only for ISAM files; if this is specified and a second I/O area is defined in the program (IOAREA2 in the FCB), read operations (GET/GETR) can be executed in overlap mode.</p> <p>Default value: depends on the BLKCTRL specification.</p> <p>For NK-ISAM, "overlapped processing" means that contiguous blocks are also read into the ISAM pool. OVERLAP=YES should be used only when reading is primarily sequential.</p>
=NO	Read operations are not to be executed in overlap mode.
=YES	Read operations are to be executed in overlap mode.
PAD=number	<p>For sequentially created ISAM files; the "padding factor" PAD specifies how much free space is to be left in each data block for subsequent extension of the file (specified as a percentage of the block size defined with BLKSIZE). PAD thus has an effect on the block splitting rate when a file is extended non-sequentially.</p> <p><math>0 \leq \text{number} \leq 99</math>; default value: PAD = 15</p> <p>The PAD specification has different effects for NK-ISAM and K-ISAM. For NK-ISAM, the block is filled at least up to the PAD limit; for K-ISAM, it is never filled above the PAD limit.</p>

- POOLLNK=name** Only for NK-ISAM files processed in ISAM pools (NK-ISAM); "name" is the "pool link name" (up to 8 characters long) which is entered in the TFT. (The ISAM pool is created by means of the CREATE-ISAM-POOL command, and the pool link name is assigned to it using ADD-ISAM-POOL-LINK.)  
The valid character set for "name" corresponds to the one for file names: letters, digits, and the special characters "#" and "@".
- RECFORM** Specifies the record format of the file "pathname" and also specifies which control characters are to be interpreted if the file is sent to a printer.  
Default value: RECFORM = (V,N)  
The record format specification is evaluated only for the access methods SAM and ISAM. UPAM processes files only on a block-by-block basis; the specification for the RECFORM operand is ignored. BTAM accepts a RECFORM specification. For the relationships between the RECFORM and RECSIZE specifications, see "RECSIZE". For information on the interpretation of the print control characters, see the PRINT command (operand SPACE).  
For tape files with RECFORM=V and CODE=EBCDIC or LABEL=(STD,n), where (n > 1), the contents of the record length and block size fields are converted internally to the D format: the value for the record/block size is represented as a decimal number. For such files, the block size must be less than 10000 bytes. During input, records with a D format are reconverted to hexadecimal format before they are transferred to the user area.  
RECFORM can also be specified as a NULL operand, in which case the RECFORM information from the catalog entry of the file will be transferred when the file is opened.
- =V** "pathname" consists of variable-length records, which means that the user must bear in mind, when programming, that each record is preceded by a 4-byte field whose first two bytes contain the record length in binary form. Bytes 3 and 4 of this field are used by the system. For input files, the record length field is specified by the system; for output files, you must do this yourself. For BTAM files, the specification RECFORM=V is treated like RECFORM=U.
- =F** "pathname" consists of fixed-length records, i.e. the user does not need to take record length and control fields into account. All records in the file have the same length, which is defined via the RECSIZE operand.



- =U "pathname" consists of records with "undefined" length. Each data block contains one, and only one record, whose length is transferred to a register. The system sets this register for input and the user program sets it for output (see the BLKSIZE operand). RECFORM=U converts the specification LABEL=(STD,3) into (STD,2).  
RECFORM=U is not permitted for ISAM files.
- =(...,N) "pathname" is not a print file and therefore contains no print control characters. It should not be printed with control character evaluation.
- =(...,M) The first data byte in each record is interpreted as a control character in EBCDIC code and the file can be printed with the specification SPACE=E in the PRINT command. For ISAM files, the ISAM index is taken into account.
- =(...,A) The first data byte in each record is interpreted as an ASA control character and the file can be printed with the specification SPACE=A in the PRINT command.
- RECSIZE Specifies the record length as a function of the specification in the RECFORM operand.  
RECSIZE can also be specified as a NULL operand, in which case the RECSIZE information from the catalog entry of the file will be taken over when the file is opened.
- =length For RECFORM=F: the record length in bytes (all records in the file are the same length).  
For RECFORM=V: The RECSIZE specification is ignored, unless an ISAM file is being read in move mode. If "length" is smaller than the record to be read, only the specified "length" of the record is transferred, and error handling is initiated.  
RECSIZE=0 means  $RECSIZE \leq BLKSIZE$ .  
For NK-ISAM files, it should be noted that overflow blocks may result if the maximum record length is fully utilized.  
For tape files, the interaction with the operands CODE and LABEL should be noted: if CODE=EBCDIC or LABEL > 1, RECSIZE must be  $\leq 9999$  (international standard).
- =reg For RECFORM=U: the operand RECSIZE must specify a general-purpose register ( $2 \leq reg \leq 12$ ) which contains the current record length for input and output. The system sets this register for input and the user program must set it for output.

- RETPD=days** Defines a retention period for the file in days, i.e. a period during which the file may be read, but not updated or erased.
- The RETPD operand is effective only if a TFT entry is created with the LINK operand and the file is then opened.
- "days" specifies the length of the retention period in days (0..32767). The date on which the retention period ends is computed by the system and noted in the EXDATE field of the catalog entry (see the FSTATUS command).
- Default value: days=0, i.e. the file may be updated or erased immediately.
- The retention period can also be controlled by the CATALOG command (see the CATALOG command for details).
- 
- SECLEV** Only for tape files; the operand SECLEV (security level) refers to the TPIGNORE entry in the JOIN entry of the user ID (see also the SHOW-USER-ATTRIBUTES command). SECLEV specifications are ignored in interactive mode. In batch mode, if you have the appropriate authorization, you can use SECLEV to specify whether error messages are to be suppressed and/or whether additional label checking is to be executed.
- =HIGH** In batch mode, error messages are sent to the system operator console. If the job is running under a user ID with TPIGNORE=YES in its JOIN entry, the operator may ignore the error messages.
- =LOW** Permissible only for the system administrator or the tape/file owner if TPIGNORE=YES is defined in the JOIN entry of the user ID: certain error messages are suppressed in batch mode.
- =(...,OPR)** The entry OPR (= overwrite protection) causes the system to execute additional label checking:
- if a file is written on a tape behind an existing file, the labels of the preceding file are checked;
  - the expiration date of the new file must not be greater than that of the preceding file.

- SHARUPD** Only for ISAM or UPAM disk files; specifies whether several jobs may concurrently open the file with an open mode other than OPEN INPUT.
- =NO** As soon as the file is opened by a job with OPEN  $\neq$  INPUT, it is locked for all other jobs. Concurrent access to the file by several jobs is possible only if the file is used as an input file by all of these jobs, i.e. it is opened with OPEN INPUT. If the file has been opened with OPEN INPUT, any attempt to open it with another open mode is rejected; default value.
- =YES** Only for ISAM and PAM files: the file can be processed concurrently by several jobs, provided SHARUPD=YES is specified in all of these jobs. With UPAM, data blocks can be protected against access by other jobs as long as processing takes place. For ISAM, these locks are set automatically by the system whenever necessary. With NK-ISAM, the shared-update processing of files must be carried out in host-specific user ISAM pools. SHARUPD=YES simultaneously activates WROUT=YES for K-ISAM files (see the WROUT operand, below).
- SPACE** Only for disk files: controls, with the primary, secondary or absolute allocation, the storage space reserved for the file. The SPACE operand is always evaluated, even if the LINK operand is not concurrently specified.
- Default value: the values defined for primary and secondary allocation during system generation
- If you request more space in the subset than is assigned to you in the JOIN entry (see the SHOW-USER-ATTRIBUTES command), the FILE command is rejected. If you are authorized to use enforcement, the system informs you with a message before your free storage space assignment is exceeded
- In the case of private disks, a partial allocation is made (whenever possible) if the request exceeds the amount of free storage space.
- The command FILE ...,SPACE=... is rejected for files which are open or for which exclusive reservations have been made by a foreign task (with SECURE-RESOURCE-ALLOCATION, for example). File protection attributes, if any, are taken into account.

In order to minimize the management overhead for the system, the following should be noted when primary and secondary allocations are defined:

- the primary allocation should match the estimated size of the file to be created;
- the secondary allocation should be between 9 and 15 PAM pages;
- for large files, the primary and secondary allocations should be multiples of the management units packet or segment (24 and 192 PAM pages, respectively).

Relationship with BLKSIZE: when defining "primary" and "secondary", the block size defined for the file should be taken into account; for  $BLKSIZE=(STD,n)$  -- where  $n \geq 2$  -the following applies:

File type	SPACE operand	
	"primary"	"secondary"
SAM	$\geq 2n$	$\geq n$
K-ISAM	$\geq 2n + 1$	$\geq n$
NK-ISAM	$\geq 2n + 2$	$\geq n$
PAM (chained I/O)	$= n$	$= n$

=primary The primary allocation or release of PAM pages, which is immediately effective.

**1 ≤ primary ≤ 50331645**: storage space allocation; "primary" is rounded up to a multiple of 3 and the corresponding number of PAM pages is allocated on the pubset or on the private disk specified in the VOLUME operand. When using the FILE command (macro) in procedures (programs), please note that each FILE command with a positive primary allocation reserves space for the file. If the primary allocation is large, this will quickly exhaust the storage space assignment for the user ID.

**-50331645 ≤ primary ≤ -1**: storage space release after rounding "primary" up to a multiple of 3.

Only the user ID from the LOGON command may be specified in order to release storage space.

The space is released from the end of the file, working backwards, as specified in the volume list (any specification in the VOLUME operand is ignored). Only "unused" units (1 unit = 3 PAM pages) are released. For ISAM files, the data and index sections cannot be released separately (see the DSPACE operand, above). In the case of files on private disks, at least 3 PAM pages remain allocated to the file, even if they are unused.

If a file has been cataloged with DESTROY=YES, all PAM pages that can be released are first overwritten with binary zeros. However, since units are not taken into account when the PAM pages are overwritten (contrary to when they are released), the file may be overwritten with binary zeros up to the end-of-file pointer. If the file resides on private disks, these disks are requested and may need to be mounted.

**primary=0**: no change to the storage space reservation; permissible for files on private disks only if the file already occupies storage space.

=(primary,secondary)

Defines the primary and secondary allocations. In contrast to the primary allocation, the secondary allocation is not immediately effective when the FILE command is entered, but only when the reserved space becomes insufficient during creation or extension of the file. The secondary allocation value is transferred to the catalog entry (field 2ND ALLO; cf. FSTATUS output).

**-50331645 ≤ primary ≤ 50331645**: see "primary", above.

SPACE=(0,secondary) defines or changes the secondary allocation and transfers the (new) value to the catalog entry. This may be specified for files or file generations on private disk only if the file already occupies storage space.

**secondary**: the secondary allocation, i.e. the number of PAM pages by which the storage space is to be extended if this should become necessary.

**1 ≤ secondary ≤ 32767**: "secondary" is transferred unchanged to the catalog entry. It is not rounded up to a multiple of 3 until the secondary allocation comes into force.

**secondary=0**: prevents dynamic expansion of the file.

=(page,number,ABS)

Only for files on private disk: absolute allocation (only permitted when specified together with VOLUME=vs<sub>n</sub>). The requested space is assigned as a contiguous storage area. If there is not enough free space on the disk, the FILE command is rejected; no partial allocation is made. Since the absolute allocation always refers to one volume, a separate FILE command must be issued for each volume. If the absolute allocation is the first space request for the file, the secondary allocation is set to 0.

**"page"** = the block number of the PAM page with which the space reservation is to start on the private disk. Since space is always allocated in units of 3 PAM pages, "page" must be  $3n + 1$  (where  $n \geq 0$ ).

The first PAM page on which storage space can be reserved on a disk depends on how the disk was formatted.

**"number"**: specifies how many PAM pages are to be reserved on the volume (number ≤ 50331645). The value must be a multiple of 3. As the capacity of a given disk depends on the disk type and how it was formatted, the user should ask the system administrator what the maximum permissible value for "number" is.

**ABS:** the keyword "ABS" identifies an absolute allocation.

**STATE=FOREIGN** For files on private volumes for which a system catalog entry is to be created (file import). For file generations, the group entry may also have to be reconstructed (with a CATALOG command/macro) before the generations can be imported. Files which are imported with STATE=FOREIGN should be exported from the catalog of their "old" owner (with ERASE, CATALOG).

**Files on private disk:** only the first volume of a file needs to be mounted when the file is opened, because the system creates the catalog entry from information in the F1 label of the private disk.

**Tape files:** The VSNs of all tapes occupied by the file must be listed in the correct order in the VOLUME operand.

A "foreign" indicator is set in the catalog entry, so that it is initially not possible to change the file attributes with a CATALOG command. This indicator is not reset until the file is opened.

If the foreign tape file has standard labels, the file attributes RECFORM, RECSIZE, BLKSIZE and CODE are transferred from the HDR2 label to the catalog entry when the file is opened. The file may be cataloged under more than one user ID; the system then ensures that the catalog entry and the label information are kept consistent.

If the foreign file has non-standard labels or no labels, you must specify the operands RECFORM, RECSIZE and BLKSIZE in the FILE command. If the file is cataloged under more than one user ID, each user is responsible for ensuring that consistency between the catalog entry and the label information is maintained.

The following prerequisites must be fulfilled for importing a foreign tape file:

- if you are not the file owner, the volume and the file must be shareable (indicators in the VOL1 and HDR1 labels):
- only the file owner may extend the file;
- if write protection is defined for the file, or if the retention period has not yet elapsed, the file owner must specify SECLEV=LOW in the FCB macro or FILE command/macro if the file is to be extended;
- if SECLEV=LOW is specified in FCB or FILE, any passwords which exist for the file are ignored.



The method for importing foreign tape files is not the same as that used for private disk files. The reason for this is that the catalog entry for a foreign disk file is unique. For foreign tape files, this uniqueness could be achieved if the user IDs of the file owners already existed in the system into which the file is to be imported. If, however, these user IDs do not exist, it is not possible to change the owner identification on the tape (a hardware restriction would cause the file to be destroyed). Even if the system administrator imports a file for an existing user ID, it cannot be guaranteed that the catalog entry will be unique, since the file can also be cataloged under another user ID.

Nevertheless, tape files with standard labels are protected just as well as disk files against conflicts between the file attributes specified in the labels and those in the catalog entry. This is achieved by the restrictions in the CATALOG command. The only risk factor is that the file owner may change the file attributes by specifying SECLEV=LOW in the FCB. For this reason, there should never be several catalog entries in one system for one file if the owner of the file also works in this system.

TAPEWR	Only for files on tape cartridges. The user can specify whether or not output is to be buffered (input is always buffered).
<u>=DEVICE-BUFFER</u>	Output is buffered by the tape controller, thus achieving a high data transfer rate; default value.
=IMMEDIATE	Output is not buffered.
TPMARK	Only for tape files without standard labels (LABEL=NO/NSTD); specifies whether tape marks are to be written. Tape files with LABEL=(STD,n) automatically receive tape marks after the labels.
<u>=YES</u>	Tape files with non-standard labels: the tape mark follows the label; default value.
	Tape files without labels: the tape mark is written at the beginning of the tape.
=NO	No tape mark is written.



TRANS	Only for tape files used as input files and not created with CODE=EBCDIC; specifies how the code of the file is to be converted during reading.
=YES	ISO 7-bit code or OWN code is converted into EBCDIC code; default value.
=NO	ISO 7-bit code is converted into 8-bit format by inserting a leading zero.
TSET	<p>Only for target tape files, together with LINK=name; TSET (tape set) establishes the connection with an existing TST (tape set table) entry or creates a new TST entry, as required (see above "Tape Set Table (TST)"). The corresponding TFT entry then points to this entry in the TST.</p> <p>TSET may be specified only for output files with standard labels. At the same time, a device type must be specified via the DEVICE operand (DEVICE=WORK is not permissible).</p> <p>When the first file of a file set is created or a file set is extended, the only permitted value for FSEQ is FSEQ=1 or FSEQ=NEW, respectively.</p> <p>If the specified name does not yet exist in any TST entry, a new TST entry with the file number 1 is created. The file set identifier set during OPEN corresponds to the identifier in the HDR1 label.</p> <p>If a TST entry with the specified name already exists, the file number is incremented by 1 with each FILE command/macro (see also the RELEASE command).</p> <p>The volume list in the TST entry is updated on the basis of the information in the current FILE command.</p> <p>The tape sequence is updated during OPEN processing.</p> <p>When a TFT entry linked to a TST entry is deleted, the TST entry is not released unless the deleted TFT entry is the last entry linked with it.</p>
=name	4-character alphanumeric name which establishes the link with the TST entry with this name or creates a new TST entry.

- `=(name,vsn)` 4-character alphanumeric name designating a TST entry. The volume serial number ("`vs`n") is the "file set identifier".
- If the TST entry does not yet contain a file set identifier, the VSN is added to the TST; otherwise, the specified VSN must match that of the TST entry. When a file is opened, the file set identifiers in the TST entry and in the HDR1 label must agree.
- `TVSN` Only for tape files used as input files; specifies which volumes are needed. If the TVSN operand is specified, the volume list in the catalog entry is ignored during file processing; only the volumes specified via TVSN are used. However, the catalog entry is not changed.
- The TVSN operand must not be specified together with the VOLUME operand.
- `=vs`n Volume serial number of the volume containing the file to be read.
- `=(vs`n,...) List specifying up to 255 VSNs of volumes which contain the file to be read.
- If "pathname" is a multivolume file, DMS assumes that the specified volume list matches the file section list in the VSEQ operand. It is up to you to ensure that the volume list begins with the volume containing the beginning of the file. If no VSEQ operand is specified, the system searches for the first file section when the file is opened and positions the tape to this section. After this, all file sections are loaded in the existing order.
- `VALLEN=length` Only for K-ISAM files; specifies the length of the value flag in the ISAM index. Value flags are evaluated block-by-block and transferred to the next higher index entry as specified in the VALPROP operand.
- VALLEN can also be specified as a NULL operand, in which case the VALLEN information from the catalog entry of the file will be taken over when the file is opened.
- Default value:      length = 0: the ISAM index does not contain a value flag.
- length ≤ 255 - KEYLEN - LOGLEN (cf. LOGLEN operand)

VALPROP	<p>Only for K-ISAM files; specifies how the value flag is to be included in the index entries (VALPROP = value propagation).</p> <p>The VALPROP operand is ignored for NK-ISAM file.</p> <p>VALPROP can also be specified as a NULL operand, in which case the VALPROP information from the catalog entry of the file will be transferred when the file is opened.</p>
= <u>MIN</u>	The lowest value of the value flag within each data or index block is included in the index entry at the next higher level; default value.
=MAX	The highest value of the value flag within a data or index block is included in the index entry at the next higher level.
VOLUME	<p>Specifies which private volumes are required for file processing. If a new file is to be created and neither DEVICE nor VOLUME is specified, the file will be processed on public volumes.</p> <p>The VOLUME operand may be omitted if STATE=FOREIGN is specified and the system parameter VSEQPRI is set.</p> <p>If private volumes are requested for processing of file generations whose group entry contains DISP=REUSE, the VOLUME specification will only be accepted until the maximum number of simultaneously existing generations is reached.</p> <p>For disk files, only the first volume specified is used. In the case of tape files, the volumes specified in the VOLUME operand are added to the volume list in the catalog entry for the "oldest" generation.</p> <p><b>Disk files</b></p> <p>If the first private disk has at least the amount of space available as is requested by the SPACE operand or if SPACE is not specified, only the first private disk specified is requested. If SPACE requests more space than is available on the first private disk, a number of private disks are requested simultaneously. If further VSNs are specified, these are added to the volume list of the catalog entry for later extensions.</p> <p>If a file is being extended, the SPACE operand must be specified at the same time with a primary allocation &gt; 0; otherwise, the specified VSNs will be ignored.</p>

## Tape files

If "pathname" is not yet cataloged, the volume list in the VOLUME operand is added to the catalog entry. A connection to a TST can be established by specifying the TSET operand at the same time. In this case, the volume list is added to the TST entry and appended to any existing list. After the file has been opened, the catalog entry is updated using the volume list in the TST entry.

If "pathname" is already cataloged, the volume list in the catalog entry is extended by adding to it the volume list in the VOLUME operand. Consequently, VSNs that already exist in the catalog entry must not be specified with VOLUME.

The first volume from the volume list is requested by default (exception: MOUNT=0). If more than one volume is to be mounted, the MOUNT operand must be used to specify how many volumes are to be mounted concurrently.

=PRIVATE	Requests a private volume at the console.
=(PRIVATE,n)	$1 \leq n \leq 9$ ; requests "n" private volumes at the console.
=vsn	The volume serial number (VSN) of the requested volume.
=(vsn,...)	A list of up to 255 different VSNs of requested volumes.


**VSEQ** Only for cataloged tape files with standard labels; the VSEQ operand permits files to be processed in sections. A file section is that part of a multivolume file which is stored on one tape. The VSNs of all volumes belonging to the file are contained in the volume list of the catalog entry. Access to a specific volume is attained via the position of the corresponding VSN in the volume list of the catalog entry.

### Example

If the VSN of the volume to be processed is in the fifth position in the volume list, the value "5" must be specified.

The VSEQ operand affects the structure of the TFT volume list:

- If VSEQ is not specified, the VSNs are arranged in the TFT in the order that is specified in the VOLUME or TVSN operand.
- If only one file section number is specified for VSEQ, all volumes as of the one identified are transferred to the TFT volume list.
- If a list of file section numbers is specified for VSEQ, only the entries in the VSEQ list are transferred to the TFT volume list.

=number	1 ≤ number ≤ 255; specifies the file section at which processing is to start.
	If "pathname" is an output file (OPEN=OUTPUT/OUTIN), VSEQ=1 must be specified.
	If "pathname" is opened or extended with OPEN EXTEND, VSEQ specifies the file section at which extension is to begin.
	For files opened with OPEN REVERSE, only one file section number may be specified; automatic tape swapping is not supported with OPEN REVERSE.
=(L(number,...))	Specifies the order in which the file sections are to be processed. This may be used only for input files, not for output files. For files opened with OPEN REVERSE, only one file section number may be specified and automatic tape swapping is not supported.
WRCHK	Write check; only for processing of disk files; specifies whether a read-after-write check is to be executed. "WRCHK" is not recorded in the catalog entry and must therefore be repeated each time before the file is opened or processed.
	A "read-after-write" check implies a check for write errors. If an unrecoverable error is detected, control is passed to the EXLST exit ERRADR. Owing to the additional disk accesses involved, this function has a markedly adverse effect on system performance.
= <u>NO</u>	No read-after-write check is executed; default value.
= <u>YES</u>	A read-after-write check is executed.
WROUT	For ISAM processing; WROUT controls the reloading of updated blocks to disk.
	Default value:
	WROUT=NO for file processing with SHARUPD=NO
	WROUT=YES for file processing with SHARUPD=YES
= <u>NO</u>	An updated block is not reloaded to disk until the contents of the buffer area in question must be replaced, or at the latest when the file is closed; default value.
	 This entry has no effect on K-ISAM files because updated blocks from these files are always reloaded to disk. WROUT=NO only works for NK-ISAM files if WRITE-IMMEDIATE=NO was specified when the ISAM pool being used was created.

=YES

Each updated block is reloaded to disk immediately, thus always ensuring the consistency of the data on disk and in virtual memory. Note, however, that this also increases the I/O rate

## **Examples**

*Example 1a:* File link name and file attributes

In program X, assume that the following specifications have been made in the file control block (FCB) for an output file:

```
FILE=X.OUTPUT  
LINK=OUTP  
FCBTYPE=SAM  
BLKSIZE=STD
```

Nothing is specified for DEVICE, which means that the system uses a public disk as the volume for this file.

Program X is called:

Fig. FILE-3 Matching the FCB to the TFT entry when a file is opened

The file is opened with the current FCB and the file is written to tape without anything in program X having to be changed.

*Example 1b: Creating a TFT entry (with NULL operands)*

```

(IN) FILE MAX.FILE, LINK=LINKIN, FCBTYPE=, RECSIZE=, RECFORM=
(IN) RDTFT LINK=LINKIN, ALL
(OUT) % LINK=LINKIN
      FILE=:W:$US123456.MAX.FILE
STATUS = INACTIVE
COMMAND = FILE
RETENT. PERIOD =NONE          SECLEV = HIGH
OVERRIDE PROTECT=NO          BYPASS = NONE
DESTOC = NONE
FCBTYPE = NULL              OPEN = NONE
RECFORM = NULL              RECSIZE = NULL
BLKSIZE = NONE              BUFOFF = NONE
KEYPOS = NONE              KEYLEN = NONE
BLKCTRL = NONE             POOLLNK = NONE
LOGLEN = NONE              VALLEN = NONE
VALPROP = NONE             DUPEKY = NONE
PAD = NONE                 OVERLAP = NONE
SHARUPD = NONE             WROUT = NONE
LABEL = NONE               TPMARK = NO
CODE = NONE                TRANS = NONE
BLIM = NONE                CHKPT = (NO, ACTIVE)
FSEQ = NONE                WRCHK = NONE
TAPEWR = NONE              CLOSMG = NONE
DEVICE = NONE              TSET NAME = NONE
VSN/DEV = PUBW02/D3480

```

*Example 2: Creating a TFT entry (file attributes defined in the FILE command)*

```
(IN) FILE MAX.FILE.4, LINK=LINKIN, RECSIZE=100, RECFORM=F, RETPD=20,
BLKCTRL=DATA, POOLLNK=ISAMPOOL
(IN) RDTFT LINK=LINKIN, ALL
% LINK=LINKIN
FILE=:W:$US123456.MAX.FILE.4
STATUS = INACTIVE
COMMAND = FILE
RETENT. PERIOD =00020 SECLEV = HIGH
OVERRIDE PROTECT=NO BYPASS = NONE
DESTOC = NONE
FCBTYPE = NONE OPEN = NONE
RECFORM = FIXED LENGTH RECSIZE = 00100
BLKSIZE = NONE BUFOFF = NONE
KEYPOS = NONE KEYLEN = NONE
BLKCTRL = DATA POOLLNK = ISAMPOOL
LOGLEN = NONE VALLEN = NONE
VALPROP = NONE DUPEKY = NONE
PAD = NONE OVERLAP = NONE
SHARUPD = NONE WROUT = NONE
LABEL = NONE TPMARK = NO
CODE = NONE TRANS = NONE
BLIM = NONE CHKPT = (NO, ACTIVE)
FSEQ = NONE WRCHK = NONE
TAPEWR = NONE CLOSMG = NONE
DEVICE = NONE TSET NAME = NONE
VSN/DEV = PUBW11/D3480
```



*Example 3: Allocating storage space on public volumes*

```

(IN) FSTAT MAX.SPACE
(OUT) % DMS0533 REQUESTED FILE NOT CATALOGED IN PUBSET 'W'. COMMAND TERMINATED
(IN) FILE MAX.SPACE
(IN) FSTAT MAX.SPACE,ALL
(OUT) 0000003 :W:$US123456.MAX.SPACE
      FCBTYP = NONE                VSNTYPE = PUB
      LASTPG = 00000000            2ND ALLO= 00003
      SHARE  = NO                  ACCESS  = WRITE
      ACL    = NO                  AUDIT   = NONE                DESTROY = NO
      CRDATE = NONE                EXDATE  = NONE                LADATE  = NONE
      RDPASS = NONE                WRPASS  = NONE                EXPASS  = NONE
      ACCESS# = 000                VERSION = 000
      LARGE  = NO                  BACKUP  = A                    MIGRATE = ALLOWED
      BLKTYE = NONE                BLKSIZE = 000000            BLKCTRL = NONE
      RECFOR = NONE                RECSIZE = 000000
      VSN/DEV/EXT = PUBW24 / D3480 / 001
      EXTCNT = 1
      :W: PUBLIC: 1 FILE RES= 3 FREE= 3 REL= 3 PAGES

(IN) COPY MAX.FILE,MAX.SPACE
(IN) FSTAT MAX.SPACE,ALL
(OUT) 0000003 :W:$US123456.MAX.SPACE
      FCBTYP = SAM                VSNTYPE = PUB
      LASTPG = 00000001            2ND ALLO= 00003
      SHARE  = NO                  ACCESS  = WRITE
      ACL    = NO                  AUDIT   = NONE                DESTROY = NO
      CRDATE = 1990-12-18          EXDATE  = 1990-12-18          LADATE  = 1990-12-18
      RDPASS = NONE                WRPASS  = NONE                EXPASS  = NONE
      ACCESS# = 001                VERSION = 001
      LARGE  = NO                  BACKUP  = A                    MIGRATE = ALLOWED
      BLKTYE = STD                 BLKSIZE = 002048            BLKCTRL = PAMKEY
      RECFOR = (V,N)              RECSIZE = 000000
      VSN/DEV/EXT = PUBW24 / D3480 / 001
      EXTCNT = 1
      :W: PUBLIC: 1 FILE RES= 3 FREE= 2 REL= 0 PAGES

(IN) FILE MAX.SPACE,SPACE=(100,20)
(IN) COPY MAX.FILE,MAX.SPACE
(IN) FSTAT MAX.SPACE,ALL
(OUT) 0000105 :W:$US123456.MAX.SPACE
      FCBTYP = SAM                VSNTYPE = PUB
      LASTPG = 00000001            2ND ALLO= 00020
      SHARE  = NO                  ACCESS  = WRITE
      ACL    = NO                  AUDIT   = NONE                DESTROY = NO
      CRDATE = 1990-12-18          EXDATE  = 1990-12-18          LADATE  = 1990-12-18
      RDPASS = NONE                WRPASS  = NONE                EXPASS  = NONE
      ACCESS# = 002                VERSION = 001
      LARGE  = NO                  BACKUP  = A                    MIGRATE = ALLOWED
      BLKTYE = STD                 BLKSIZE = 002048            BLKCTRL = PAMKEY
      RECFOR = (V,N)              RECSIZE = 000000
      VSN/DEV/EXT = PUBW24 / D3480 / 002
      EXTCNT = 2
      :W: PUBLIC: 1 FILE RES= 105 FREE= 104 REL= 102 PAGES

```

```
(IN) FILE MAX.SPACE,SPACE=-150
(IN) FSTAT MAX.SPACE,STANDARD
(OUT) 0000003 :W:$US123456.MAX.SPACE
      FCBTYPE = SAM          VSNTYPE = PUB
      LASTPG  = 00000001     2ND ALLO= 00020
:W: PUBLIC: 1 FILE. RES= 3, FREE= 2, REL= 0 PAGES

(IN) FILE MAX.PRIV.6,DEVICE=D3480,VOLUME=WORK02,SPACE=(100,20)
(IN) FSTAT MAX.PRIV.6,STANDARD
(OUT) 0000102*:W:$US123456.MAX.PRIV.6
      FCBTYPE = SAM          VSNTYPE = PUB
      LASTPG  = 00000000     2ND ALLO= 00020
:W: PRIVATE: 1 FILE. RES= 102, FREE= 102, REL= 99 PAGES

(IN) FILE MAX.PRIV.6,SPACE=-150
(IN) FSTAT MAX.PRIV.6
(OUT) 0000003*:W:$US123456.MAX.PRIV.6
      :W: PRIVATE: 1 FILE. RES= 3, FREE= 3, REL= 0 PAGES

(IN) FILE MAX.FILE.6,SPACE=(100,20)
(IN) FSTAT MAX.FILE.6
(OUT) 0000102 :W:$US123456.MAX.FILE.6
      :W: PUBLIC: 1 FILE. RES= 102, FREE= 102, REL= 102PAGES

(IN) FILE MAX.FILE.6,SPACE=-150
(IN) FSTAT MAX.FILE.6
(OUT) 0000000 :W:$US123456.MAX.FILE.6
      :W: PUBLIC: 1 FILE. RES= 0, FREE= 0, REL= 0PAGES
```

*Example 4: VSEQ operand=*

- /FILE A,VSEQ=2,VOLUME=V3,LINK=linkname  
Volume list in the catalog before execution of the FILE command: (V1,V2)  
Volume list in the catalog after execution of the FILE command: (V1,V2,V3)  
TFT volume list: (V2,V3)
- /FILE A,VSEQ=(L=(1,2)),LINK=linkname  
Volume list in the catalog: (V1,V2,V3,V4)  
TFT volume list: (V1,V2)

*Example 5a: TSET operand*

```
/FILE A,VOLUME=V1,TSET=X,FSEQ=NEW,DEVICE=T1600,LINK=linkname1  
/FILE B,VOLUME=V2,TSET=X,FSEQ=NEW,DEVICE=T1600,LINK=linkname2
```

The VSNs V1 and V2 are entered in the TST entry in accordance with the specifications in the TSET operand. The part of the catalog entry containing the list of volume serial numbers is structured as specified in the TST entry. If, in this example, the TST entry is created when the first FILE command is executed, the VSN list for file A contains the value (V1) and the list for file B contains (V1,V2), rather than only (V2), which would have been the case if the TSET operand had not been specified.

*Example 5b: TSET operand*

```
/FILE A, LINK=A, VOLUME=V1, TSET=X, FSEQ=NEW, DEVICE=T1600
```

After execution of this FILE command, the VSN V1 is inserted in the device table of the TST entry "X" and cataloged for file A.

```
/FILE B, LINK=B, TSET=X, FSEQ=NEW, DEVICE=T1600
```

After execution of this FILE command, the VSN V1 is also cataloged for file B.

```
OPEN A
.
.
.
FEOV
.
.
.
CLOSE A, LEAVE
```

File A is created on the tape with VSN V1 and, due to the FEOV macro, extended on the next tape with the VSN V2, for example. V2 is then inserted in the device table for the TST entry "X" and added to the catalog entry for file A (see also EOV processing). The tape sequence for file A thus contains the VSNs (V1,V2) and the pointer to the current tape in the TST entry "X" points to V2.

```
OPEN B
.
.
.
CLOSE B
```

Since the volume/tape pointer of TST entry "X" points to the tape with VSN V2, file B is written to this tape and the tape sequence for file B is thus (V2) instead of (V1).

*Example 6: Tape dump with the PERCON utility routine*

The pubset file MAX.TAPE.FILE is to be copied to tape. For this, the "copy" MAX.TAPE.FILE.1 is first cataloged with a FILE command and the file MAX.TAPE.FILE is then copied.

The PERCON utility routine shows, in the SYSOUT log, the contents of the new tape file both in plain text and in hexadecimal notation. The SYSLST log (file MAX.PERCON.2) shows only the printable text.

With the aid of the tables in the appendix, it is easy to see the structure of the labels. The tape marks cannot be represented in printable form in the SYSLST log, nor can the block size and record length fields.

(\*) in the log identifies PERCON statements

```
(IN) FILE MAX.TAPE.FILE.1,DEVICE=TAPE,VOLUME=D5260A
(IN) COPY MAX.TAPE.FILE,CMH.TAPE.FILE.1
(OUT) % DMS0DE3 VOL D5260A FOR FILE :W:$US123456.MAX.TAPE.FILE.1 IS
      MOUNTED ON DEVICE T0
(IN) EXEC $PERCON
      % BLS0500 PROGRAM 'PERCON', VERSION 'V2.2AB4' OF '87-06-11' LOADED.
      % PER0000 PERCON BS2000 VERSION 2.2A00
(IN) volin volume=d5260a _____(*)
(OUT) % DMS0DF6 INPUT TAPE WITH VSN D5260A HAS STD LABELS, BUT SHOULD BE
      TREATED AS NSTD OR NO REPLY (0=EXIT; C=CONTINUE)?
(IN) C
(OUT) % DMS0DE3 VOL D5260A FOR FILE :W:$US123456.PERCON.TPWORK.D.3686.VS0001
      IS MOUNTED ON DEVICE T0
(IN) volout device=display } _____(*)
(IN) edit } _____(*)
(OUT) TMCNT: 000          BLOCK: 00000001
      V O L 1 D 5 2 6 0 A
(00000) E5D6D3F1C4F5F2F6 F0C1404040404040 4040404040404040 4040404040404040
(00032) 4040404040404040 4040404040404040 4040404040404040 4040404040404040
(00064) 4040404040404040 40404040404040F1

      TMCNT: 000          BLOCK: 00000002
      H D R 1 C M H .   B A N D . D A T   E I . 1   D 5 2   6 0 A 0 0 0 1 0
(00000) C8C4D9F1C3D4C84B C2C1D5C44BC4C1E3 C5C94BF140C4F5F2 F6F0C1F0F0F0F1F0
      0 0 1 0 0 0 1 0 0 0 8 7 3 2 4 8 7 3 2 4 0 0 0 0 0 0 B S 2 0
(00032) F0F0F1F0F0F0F1F0 F040F8F7F3F2F440 F8F7F3F2F440F0F0 F0F0F0F0C2E2F2F0
      0 0
(00064) F0F0404040404040 4040404040404040

      TMCNT: 000          BLOCK: 00000003
      H D R 2 V 8 0 0 0 1 0 2 0 4 8 0 /
(00000) C8C4D9F2E5F8F0F0 F0F1F0F2F0F4F840 F061404040404040 4040404040404040
      0 4
(00032) 4040404040404040 4040404040403180 0A33F0F440404040 4040404040404040
(00064) 4040404040404040 4040404040404040

      TMCNT: 000          BLOCK: 00000004
      H D R 3 Q M 2 8 6 0 1 8 C M H .   B A N D . D A T   E I . 1
(00000) C8C4D9F3D8D4F2F8 F6F0F1F8C3D4C84B C2C1D5C44BC4C1E3 C5C94BF140404040
(00032) 4040404040404040 4040404040404040 4040404040404040 0000000000000000
      0 0
(00064) 00000000F0F04040 4040404040404040

*** TAPE MARK ***
```

```

TMCNT: 001          BLOCK: 00000001
(00000) 31800A3301000001 00000100002D0000 002D404000114040 C48985A24089A2A4
          e i n e           k l e i n e           T e s t d a t e i .
(00032) 408589958500A50 4092938589958500 0E4040E385A3A484 81A385894B0000000
(00064) 0000000000000000 0000000000000000 0000000000000000 0000000000000000
          ***          00061 IDENTICAL LINES SUPPRESSED          ***

```

```
(02048) 0000000000000000 0000000000000000
```

\*\*\* TAPE MARK \*\*\*

```

TMCNT: 002          BLOCK: 00000001
(00000) E O F 1 C M H . B A N D . D A T E I . 1 D 5 2 6 0 A 0 0 0 1 0
          C5D6C6F1C3D4C84B C2C1D5C44BC4C1E3 C5C94BF140C4F5F2 F6F0C1F0F0F0F1F0
          0 0 1 0 0 0 1 0 0 8 7 3 2 4 8 7 3 2 4 0 0 0 0 0 1 B S 2 0
(00032) F0F0F1F0F0F0F1F0 F040F8F7F3F2F440 F8F7F3F2F440F0F0 F0F0F0F1C2E2F2F0
          0 0
(00064) F0F0404040404040 4040404040404040

```

```

TMCNT: 002          BLOCK: 00000002
(00000) E O F 2 V 8 0 0 0 1 0 2 0 4 8 0 /
          C5D6C6F2E5F8F0F0 F0F1F0F2F0F4F840 F061404040404040 4040404040404040
          0 4
(00032) 4040404040404040 4040404040403180 0A33F0F440404040 4040404040404040
(00064) 4040404040404040 4040404040404040

```

```

TMCNT: 002          BLOCK: 00000003
(00000) E O F 3 Q M 2 8 6 0 1 8 C M H . B A N D . D A T E I . 1
          C5D6C6F3D8D4F2F8 F6F0F1F8C3D4C84B C2C1D5C44BC4C1E3 C5C94BF140404040
(00032) 4040404040404040 4040404040404040 4040404040404040 0000000000000000
          0 0
(00064) 00000000F0F04040 4040404040404040

```

\*\*\* TAPE MARK \*\*\*

\*\*\* TAPE MARK \*\*\*

```

(IN)      filout filnam=cmh.percon.2 (*)
(OUT)    % PER0033 LINK NAME 'PCOUT' MULTIPLE USED. LAST STATEMENT IS VALID
(IN)      edit }
(IN)      end   } (*)
(OUT)    % PER0029 NUMBER OF HANDLED BLOCKS FOR LINK=PCIN: 16
          % PER0030 NUMBER OF HANDLED RECORDS FOR LINK=PCOUT (FILE=MAX.PERCON.2): 8
          % PER0031 PERCON NORMAL END

```

PERCON log: fileout

```

VOL1D5260A
HDR1MAX.TAPE.FILE.1 D5260A00010001000100 87324 87324 000000BS2000
HDR2V8000102048 0/ 04
HDR3US123456MAX.TAPE.FILE.1 00
          This is a small test file.
EOF1MAX.TAPE.FILE.1 D5260A00010001000100 87324 87324 000001BS2000
EOF2V8000102048 0/ 04
EOF3US123456MAX.TAPE.FILE.1 00

```

1

*Example 7: Creating a file set*

The first file of the file set is created as a copy of a disk file (with no FSEQ specification).

```
(IN) FILE MAX.TAPE.FILE.1, DEVICE=TAPE, VOLUME=D5260A
(IN) COPY MAX.TAPE.FILE, CMH.TAPE.FILE.1
```

The second file is created with the EDT file editor (LINK=EDTSAM). When EDT is terminated, the DMS issues the messages DMS0DE3 and DMS0DE7:

```
(IN) FILE MAX.BAND.MF.DAT1, DEVICE=TAPE, VOLUME=D5260A, FSEQ=NEW, LINK=EDTSAM, BLKSIZE=30

(OUT) %DMS0DE3 TAPE D5260A FOR FILE MAX.TAPE.MF.DAT1 ASSIGNED TO DEVICE A0
(OUT) %DMS0DE7 TAPE-FILE MAX.TAPE.MF.DAT1 WITH LINK-NAME EDTSAM CLOSED,
      BLOCK COUNT = 001
```

The third file MAX.BAND.MF.DAT2 is also created with the EDT:

```
(IN) FILE MAX.TAPE.MF.DAT2, DEVICE=TAPE, VOLUME=D5260A, FSEQ=NEW, LINK=EDTSAM, BLKSIZE=30

(OUT) %DMS0DE3 TAPE D5260A FOR FILE MAX.TAPE.MF.DAT2 ASSIGNED TO DEVICE A0
(OUT) %DMS0DE7 TAPE-FILE MAX.TAPE.MF.DAT2 WITH LINK-NAME EDTSAM CLOSED,
      BLOCK COUNT = 001
```

The last file in the set is also created with the EDT:

```
(IN) FILE MAX.TAPE.MF.DAT3, LINK=EDTSAM, DEVICE=TAPE, VOLUME=D5260A, FSEQ=NEW, BLKSIZE=20, -
(IN) RECFORM=F, RECSIZE=10, RETPD=2

(OUT) %DMS0DE3 BAND D5260A FOR FILE MAX.TAPE.MF.DAT3 ASSIGNED TO DEVICE A0
(OUT) %DMS0DE7 TAPE-FILE MAX.TAPE.MF.DAT3 WITH LINK-NAME EDTSAM CLOSED,
      BLOCK COUNT = 002
```

## 7a) PERCON tape dump

A tape dump of tape D5260A is written to a file using the PERCON utility routine. The log looks like this: The non-printable characters (e.g. in the block size and record length fields) are replaced in the log by "\*\*\*".

```

VOL1D5260A
HDR1MAX.TAPE.FILE.1 D5260A00010001000100 87324 87324 000000BS2000
HDR2V8000102048 0/ ****04
HDR3US123456MAX.TAPE.FILE.1 00
*** * * * * * This is a * small * test file.
EOF1MAX.TAPE.FILE.1 D5260A00010001000100 87324 87324 000001BS2000
EOF2V8000102048 0/ ****04
EOF3US123456MAX.TAPE.FILE.1 00
HDR1MAX.BAND.MF.DAT1 D5260A00010002000100 87324 87324 000000BS2000
HDR2V000300003040/ 04
HDR3US123456MAX.TAPE.MF.DAT1 00
* * A * SHORT TEST:
* * FILE FSEQ1
EOF1MAX.BAND.MF.DAT1 D5260A00010002000100 87324 87324 000002BS2000
EOF2V000300003040/ 04
EOF3US123456MAX.TAPE.MF.DAT1 00
HDR1MAX.BAND.MF.DAT2 D5260A00010003000100 87324 87324 000000BS2000
HDR2V000300003040/ 04
HDR3US123456MAX.TAPE.MF.DAT2 00
* * A * SHORT TEST:
* * FILE FSEQ1
EOF1MAX.TAPE.MF.DAT2 D5260A00010003000100 87324 87324 000002BS2000
EOF2V000300003040/ 04
EOF3US123456MAX.TAPE.MF.DAT2 00
HDR1MAX.BAND.MF.DAT3 D5260A00010004000100 87324 87326 000000BS2000
HDR2F000200001040/ 00
HDR3US123456MAX.TAPE.MF.DAT3 00
ANOTHER SHORT TEST
OF FSEQ
EOF1MAX.TAPE.MF.DAT3 D5260A00010004000100 87324 87326 000002BS2000
EOF2F000200001040/ 00
EOF3US123456MAX.TAPE.MF.DAT3 00

```

1



# FSTATUS Request catalog information

Application groups:

File processing (page 26 ff.)

Device and volume reservation (page 29)

## Command description

The FSTATUS command serves to output information relating to catalog entries for files, file generations and file generation groups. This may consist simply of a list of names of files, etc. matching the criteria specified in the FSTATUS command, or of detailed information from the catalog entry. The user can obtain information about all files under his/her own user ID and about all shareable files owned by other users. Temporary files can only be addressed by their special prefix.

If the FSTATUS command is entered without operands, the reserved space and the path names of all files under the user's ID are listed in alphabetical order. FSTATUS HELP causes the syntax of the FSTATUS command to be output to SYSOUT (only effective in interactive mode). The FSTATUS output can be interrupted by means of the BREAK function (K2 key).

Date entry format for the operands CRDATE, EXDATE and LADATE:

The date can be entered as an absolute or a relative value.

- Absolute date entry:  
The date must be entered in the form YYYY-MM-DD, where YYYY is the year, MM the month and DD the day. The century may be omitted when entering the year, as can the leading zeros when entering the month or day.  
The entry YYMMDD is also permitted.
- Relative date entry:  
The date can also be entered in days, measured from the current day. It then has the form "-n" for the past and "+n" for the future, where "n" is the number of days. Leading zeros may be omitted; the signs, however, are mandatory.  
The following predefined concepts may be used for referring to short time spans: Y[ESTERDAY] (where n = -1), T[ODAY] (n = +0/-0) and TOM[ORROW] (n = +1).

Format and operand description

Operation	Operands
$\left\{ \begin{array}{l} \text{FSTATUS} \\ \text{FSTAT} \\ \text{FS} \end{array} \right\}$	$\left[ \begin{array}{l} \left\{ \text{pathname1} \right\} \\ \left\{ \text{prefix} \right\} \end{array} \right]$ $\left[ \left\{ \left[ \left[ \text{STANDARD} \right] \left[ \text{CATALOG} \right] \left[ \text{TRAITS} \right] \left[ \text{PASSWORD} \right] \right] \right\} \right]$ $\left[ \left\{ \left[ \text{ALL} \right] \right\} \right]$ $\left[ \text{RESERVED} \right]$ $\left[ \text{ACCESS} = \left\{ \begin{array}{l} \text{READ} \\ \text{WRITE} \end{array} \right\} \right]$ $\left[ \text{ACL} = \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \right]$ $\left[ \text{BACKUP} = \left\{ \begin{array}{l} \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \\ \text{E} \end{array} \right\} \right]$ $\left( \left\{ \begin{array}{l} \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \\ \text{E} \end{array} \right\}, \dots \right)$ $\left[ \text{BASIC-ACL} = \left\{ \begin{array}{l} \text{NONE} \\ \left\{ \text{OWNER} = \text{access-rights} \right\} \\ \left\{ \text{GROUP} = \text{access-rights} \right\} \left[ \left\{ \dots \right\} \right] \\ \left\{ \text{OTHERS} = \text{access-rights} \right\} \end{array} \right\} \right]$

Operation	Operands
{ FSTATUS FSTAT FS (cont.)	[ , BLKCTRL= { NONE PAMKEY DATA NO ( { NONE PAMKEY } , ... ) { DATA NO }           } ]
	[ , CRDATE= { date (date[ , ]) ( , date) (date1 , date2)           } ]
	[ , EXDATE= { date (date[ , ]) ( , date) (date1 , date2)           } ]
	[ , EXTENTS= { number (number[ , ]) ( , number) (number1 , number2)           } ]
	[ , FCBTYPE= { ISAM BTAM PAM SAM NONE ( { ISAM BTAM PAM SAM NONE } , ... )           } ]

Operation	Operands
{FSTATUS} {FSTAT} {FS (cont.)	$[ , \text{FREESIZE} = \left\{ \begin{array}{l} \text{SIZE} \\ \text{number} \\ (\text{number} [, ,]) \\ (, \text{number}) \\ (\text{number1}, \text{number2}) \end{array} \right\} ]$ $[ , \text{FROM} = \left\{ \begin{array}{l} \text{CATALOG} \\ \text{LOCAL-PVS} \end{array} \right\} \\ (\text{vsn}, \text{device}) ]$ $[ , \text{GEN} = \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} ]$ $[ , \text{LADATE} = \left\{ \begin{array}{l} \text{date} \\ (\text{date} [, ,]) \\ (, \text{date}) \\ (\text{date1}, \text{date2}) \end{array} \right\} ]$ $\left\{ \begin{array}{l} \text{value} \\ (\text{value} [, ,]) \end{array} \right\}$ $[ , \text{LIST} = \left\{ \begin{array}{l} \left\{ \begin{array}{l} (\text{SYSOUT}) \\ (\text{SO}) \\ (\text{SYSLST}) \\ (\text{SL}) \\ (\text{PRINT}) \\ (\text{PR}) \end{array} \right\} \\ \text{pathname2} \end{array} \right\} ]$ $\left\{ \begin{array}{l} \left\{ \begin{array}{l} (\text{SYSOUT}) \\ (\text{SYSLST}) \\ (\text{PRINT}) \\ \text{pathname2} \end{array} \right\} , \left\{ \begin{array}{l} \text{STANDARD} \\ \text{FILENAM} \end{array} \right\} \end{array} \right\}$ $[ , \text{MIGRATE} = \left\{ \begin{array}{l} \text{ALLOWED} \\ \text{INHIBIT} \\ \left\{ \begin{array}{l} \text{ALLOWED} \\ \text{INHIBIT} \end{array} \right\} , \dots \end{array} \right\} ]$

Operation	Operands
{ FSTATUS FSTAT FS (cont.) }	<pre>           [ , PASS= {                     RDPASS                     WRPASS                     EXPASS                     NONE                   } ]           [ , PASS= {                     {                       RDPASS                       WRPASS                       EXPASS                       NONE                     } , ...                   } ]           [ , SHARE= {                     NO                     SPECIAL                     YES                   } ]           [ , SIZE= {                     FREESIZE                     number                     (number [, ,])                     ( , number)                     (number1 , number2)                   } ]           [ , SORT= {                     FILENAM                     NO                   } ]           [ , STATE= {                     NOCLOS                     PCLOSE                   } ]           [ , STORAGE-LEVEL = {                     S0                     S1                     S2                     {                       {                         S0                         S1                         S2                       } , ...                     }                   } ]           [ , SUPPORT= {                     PUBLIC                     PRDISC                     TAPE                     {                       {                         PUBLIC                         PRDISC                         TAPE                       } , ...                     }                   } ]           [ , TYPE=FGG ]           [ , VOLUME=vsn ]           [ , VTOC= {                     YES                     NO                   } ]         </pre>

**Positional operands**

pathname1	stands for [:catid:][userid][filename]  File selection operand; "pathname" determines from which files information is to be requested. Temporary files are not included. You can use wildcards in "catid" and "filename" (see page 46). "pathname" must not be longer than 80 characters.
catid	Catalog ID of the pubset containing the file. Default value: the catalog ID assigned to the user ID (JOIN entry).
userid	User ID assigned to the file. Default value: the user ID from the LOGON command.

**Multiprocessor systems** (see also "MSCF" manual)

Any "catid" or "userid" may be specified. Three different cases must be distinguished:

- Neither "catid" nor "userid" is specified:  
All file entries relating to the job-specific user ID which are contained in the catalog whose ID is assigned to that user ID are displayed.
- "catid" is specified, "userid" is not specified:  
All file entries relating to the job-specific user ID which are contained in the catalog with the specified "catid" are displayed. In this case, the command has the format: FSTAT :catid:
- Only "userid" is specified:  
The file entries relating to the specified "userid" and having the default catalog ID for that "userid" are displayed.

## filename

Fully or partially qualified name of a permanent or temporary file, of a file generation, or file generation group. The list below shows the possible specifications and their meanings:

pathname1	Meaning
:catid:	Information about all of the user's files cataloged under "catid".
:catid: \$userid.	Own user ID: information about all files cataloged with "catid" under the user's ID. Foreign user ID: information on all files which are on the specified pubset ("catid") and for which the user has access rights. Temporary files are not included.
\$userid. filename	"filename" fully or partially qualified; own user ID: information about all files on the default pubset that are addressed by "filename"; temporary files are not included. Foreign user ID: information about the files addressed by "filename" if the user has access rights for those files.
:catid: \$userid. filename	"filename" fully or partially qualified; own user ID: information about all files on the specified pubset that are addressed by "filename"; temporary files are not included. Foreign user ID: information about the files addressed by "filename" on the specified pubset if the user has access rights for those files.

## prefix

File selection operand; information about all temporary files for the current job can be requested by specifying the special prefix character for temporary files only (# or @, defined at system generation time). However, in the SYSOUT or SYSLST log, the file name plus prefix specified by the user is replaced by the file name used internally by the system, whereby the prefix is replaced by the character string "S.nnn.tsn".

**Keyword operands (in alphabetical order)**

ACCESS	File selection operand; supplies information about the files/file generations for which read-only or read/write access is permitted.
=READ	Supplies information about files for which only read access is permitted.
=WRITE	Supplies information about files for which both read and write access are permitted.
ACL	File selection operand; the file is selected on the basis of the ACL entry.
=N[O]	Files that have no ACL entry are to be selected.
=Y[ES]	Files that have an ACL entry are to be selected.
ALL	Information selection operand; the information from STANDARD, CATALOG and TRAITS is combined. The table "Meaning of output fields", which starts on page 284 lists the output fields alphabetically, and gives details on their possible values.
BACKUP	File selection operand; selects files on the basis of the ARCHIVE or HSMS save level specified. A number of save levels may also be specified in list form, in which case all files and FGGs which satisfy one of these conditions are included (logical OR operation).
=A	Selects files/FGGs with the attribute BACKUP=A.
=B	Selects files/FGGs with the attribute BACKUP=B.
=C	Selects files/FGGs with the attribute BACKUP=C.
=D	Selects files/FGGs with the attribute BACKUP=D.
=E	Selects files/FGGs with the attribute BACKUP=E.
BASIC-ACL	File selection operand; selects files in accordance with the BASIC-ACL entry. Specifications within a user class are combined by means of a logical OR operation (see also the "DMS Introductory Guide and Command Interface" [8]).
=NONE	Files that have no BASIC-ACL entry are to be selected.



- =YES Files that have a BASIC-ACL entry are to be selected.
- =(...) Files are to be selected on the basis of individual specified access rights.
- OWNER= Files for which the owner has at least one of the specified access rights are to be selected.
- GROUP= Files for which members of the user's group have at least one of the specified access rights are to be selected.
- OTHERS= Files files for which members of other groups have at least one of the specified access rights are to be selected.

#### access-rights

mean:

$$\left[ \begin{array}{l} \text{NO-ACCESS} \\ \left( \left[ \text{R[EAD]} = \left\{ \begin{array}{l} \text{[NO]} \\ \text{[YES]} \end{array} \right\} \right], \left[ \text{W[RITE]} = \left\{ \begin{array}{l} \text{[NO]} \\ \text{[YES]} \end{array} \right\} \right], \left[ \text{[E]X[EC]} = \left\{ \begin{array}{l} \text{[NO]} \\ \text{[YES]} \end{array} \right\} \right] \right) \end{array} \right]$$

**NO-ACCESS** No access authorization;

**READ** read access;  
 =NO Not allowed;  
 =YES Allowed;

**WRITE** Write access  
 =NO Not allowed;  
 =YES Allowed;

**EXEC** Execute access  
 =NO Not allowed;  
 =YES Allowed;

**BLKCTRL** File selection operand; selects on the basis of file type, depending on the block format with which the file was stored (defined via the BLKCTRL operand of the FILE command).

=NONE The BLKCTRL type is not to be used as selection criterion.

=PAMKEY Files created with BLKCTRL=PAMKEY - i.e. files which have a separate PAM key - are to be selected.

=DATA Files created with BLKCTRL=DATA - i.e. whose block control information is kept in a block control field at the start of each data block - are to be selected.

=NO Files with no block control information are to be selected.

CATALOG Information selection operand; outputs information about the currently occupied storage space and about the attributes of files, file generations and file generation groups defined using CATALOG commands (for more details, refer to table "Meaning of the output fields, starting on page 284).

The following catalog entry fields are output for each selected file:

Output field	Meaning
ACCESS	Permitted access (read/write)
ACCESS#	Access counter
ACL	ACL entry exists: yes/no
AUDIT	Monitoring by system exit routines
BACKUP	Backup/save level for ARCHIVE
CRDATE	Date of last write access
DESTROY	Automatic data destruction: yes/no
EXDATE	Expiration date (date as of which write access is permitted)
EXPASS	Execute password exists: yes/no
GROUP	Access rights for your user group
LADATE	Last access date
LARGE	Complete ARCHIVE backup: yes/no
MIGRATE	Migration permitted (yes/no)
OTHERS	Access rights for other user groups
OWNER	Access rights for your user group
RDPASS	Read password exists: yes/no
SHARE	Shareability: yes/no
VERSION	Internal ARCHIVE attribute
WRPASS	Write password exists: yes/no

This is followed by total lines indicating the storage space occupied by the listed files, divided into groups according to volume type (public/private/tape). File generations are not included unless GEN=YES is specified (see the description for "STANDARD").

CRDATE File selection operand; selects files on the basis of their creation dates or the dates of their last write accesses. Area data includes the specified limit values in each case. The format of the date specification is described in the command description above. Dates which lie in the future are meaningless for CRDATE.

=date	Selects files last accessed for writing on the specified date.
=(date[,])	Selects files created or updated between the specified date and the current date.
=(,date)	Selects files last accessed for writing before the specified date.
=(date1,date2)	Selects files last accessed for writing during the specified period (date1 < date2).
EXDATE	File selection operand; selects files based on their expiration date, i.e. the date on which a defined retention period expires. Future dates indicate that the retention period for the respective file is still in effect. For the format of the date specification, see the command description above.
=date	Selects files which contain precisely the date specified in the EXDATE field.
=(date[,])	Selects files whose retention period expires on or after the date specified in the EXDATE field.
=(,date)	Selects files whose retention period expires on or before the specified date.
=(date1,date2)	Selects files whose expiration date lies within the specified period (date1 < date2).
EXTENTS	File selection operand; selects files on the basis of the number of extents they occupy. These are counted in the EXTCNT field of the catalog entry.
=number	$0 \leq \text{number} \leq 65535$ ; selects files with precisely the specified number of extents.
=(number[,])	Selects files with at least the specified number of extents.
=(,number)	Selects files with no more than the specified number of extents.
=(number1,number2)	Selects files with a number of extents within the specified range (number1 < number2).

FCBTYPE	File selection operand; selects files on the basis of the access method with which they were created. If a list of access methods is specified, the system links them by means of a logical OR operation and provides information about all files which satisfy one of the specified conditions.
=NONE	Selects files which are cataloged but contain no data, i.e. files which have not yet been opened or files which have been erased by means of /ERASE...,SPACE.
=ISAM	Returns information about ISAM files.
=BTAM	Returns information about BTAM files.
=SAM	Returns information about SAM files.
=PAM	Returns information about PAM files.
FREESIZE	Selects files and file generation groups on the basis of the size of their free (= reserved but unused) space.
=SIZE	Selects files for which the number of free PAM pages is equal to the number of reserved pages.
=number	$0 \leq \text{number} \leq 16777215$ ; selects files with precisely the specified number of free PAM pages.
=(number[,])	Selects files with at least the specified number of free PAM pages.
=(,number)	Selects files with no more than the specified number of free PAM pages.
=(number1,number2)	Returns information about files whose number of free PAM pages lies within the specified range ( $\text{number1} < \text{number2}$ ).
FROM	Defines the source for the information that is to be output.
=CATALOG	The output information is to be derived from the catalog of the default pubset for the user ID, i.e. the catalog with the default catalog ID (JOIN entry).
=LOCAL-PVS	The output information is to be derived from the system catalogs of all selected local pubsets.

=(vs<sub>n</sub>,device) The output information is to be derived from the directory of the private disk designated by "vs<sub>n</sub>". The device type ("device") of this private disk must be specified. Permissible values for "device" can be found in the device table in the appendix. The operands VOLUME, SUPPORT and VTOC must not be specified.

GEN Specifies whether or not information about file generations is to be returned.

Default value: GEN=NO

=NO }  
 =YES }

The following table shows the interaction of the GEN operand with the operand TYPE=FGG.

Operand combinations in the current FSTATUS command			FGGs	Information about	
TYPE=FGG	GEN=YES	GEN=NO		file genera- tions	files
x	x		*	*	-
x		x	*	-	-
	x		*	*	*
		x	*	-	*

- x Specification in the FSTATUS command
- \* Included in the command output
- Not included in the command output

The specification GEN=YES is effective only if no "filename" is included in "pathname1".

LADATE Selects files, file generations and FGGs on the basis of the date of the last access to the file. If the fields CRDATE and LADATE contain the same date, the last access was a write access; if CRDATE < LADATE the last access was only a read access. For the format of date specification see the command description above. Future dates such as TOMORROW are meaningless in this case.

=date Selects files whose field LADATE contains precisely the specified date.

=(date[,])           Selects files which have been accessed since the specified date.  
 =(,date)            Selects files which have not been accessed since the specified date.  
 =(date1,date2)     Selects files last accessed during the specified period (date1 < date2).

L[AST]P[AGE]        Selects files on the basis of the number of written PAM blocks (tape files: number of blocks according to the block counter BLKCNT);  $0 \leq \text{value} \leq 16777215$ .

=value               Selects files with precisely the specified number of written PAM blocks.

=(value,)            Selects files with at least the specified number of written PAM blocks.

=(,value)            Selects files with no more than the specified number of written PAM blocks.

=(value, value)     Selects files whose number of written PAM blocks lies within the specified range.

LIST                 Specifies the output medium for current FSTATUS processing. The requested information is output in the form of a table.

The operands STANDARD, CATALOG, TRAITS, PASSWORD, ALL and RESERVED may not be specified together unless the FSTATUS output is sent to SYSOUT.

= { ( SYSOUT ) }  
 { (SO) }

In interactive mode, the FSTATUS output is sent to the user's terminal and is determined by the operands STANDARD, CATALOG, TRAITS, ALL, RESERVED and PASSWORD; default value.

= { (SYSLST) }  
 { (SL) }

The FSTATUS output is sent to the system file SYSLST; the type of output is determined by the additional specification STANDARD or FILENAM (see below). SYSLST can be assigned (beforehand) to a temporary or permanent file by means of a SYSDFILE command.

= { (PRINT) }  
 { (PR) }

The FSTATUS output is sent directly to a printer. The type of output is determined by the additional specification STANDARD or FILENAM (see below).

=pathname2 The FSTATUS output is written to the file specified here by "pathname2". The type of output is determined by the additional specification STANDARD or FILENAM (see below). "pathname2" must not be identical to "pathname1".

pathname2 stands for :catid:\$userid.filename

:catid: Catalog ID  
 userid User ID  
 filename Fully-qualified name of a file or file generation

=(,STANDARD)

The FSTATUS command generates an edited list with the following information:

Output field	Meaning
FILENAME	File/path name
PAM PAGES	Reserved storage space
FREE PAGES	Reserved but unused storage space
SECOND ALLOC.	Secondary allocation
FCBTYPE	Access method at file creation time
SHARE	Shareable: yes/no
ACCESS	Permissible access type: read/write
PASSWORDS	Passwords exist: yes/no
BKL	Backup/save level
#EXT	Total number of extents for the file
VOLUME	Volume serial number of the volume

=(,FILENAM) Generates an unedited list with the names of the files, file generations and file generation groups in question (the first byte of each output record is X'40').

MIGRATE Only for files and FGGs on disk; files are selected on the basis of the migration option.

=ALLOWED Selects files and file generation groups for which migration is permitted.

=INHIBIT Selects files and file generation groups for which migration is not permitted.

<b>PASS</b>	Selects files and file generation groups depending on the password defined in the CATALOG command. If more than one password type is specified in the form of a list, the system logically ORs them and provides information on all files which satisfy one of the specified conditions.
<b>=NONE</b>	Selects files for which there is no password protection.
<b>=RDPASS</b>	Returns information on which files are protected by read passwords. The actual passwords are not shown.
<b>=WRPASS</b>	Returns information about files which are protected by write passwords. The actual passwords are not shown.
<b>=EXPASS</b>	Returns information about files which are protected by execute passwords. The actual passwords are not shown.
<b>PASSWORD</b>	Returns information about the type of password protecting a file or file generation group. The actual passwords are not shown. If you have forgotten your password, you must ask the system administrator for assistance.  The total lines for the amount of storage space occupied are also output (see the description of "STANDARD" and also "Meaning of output fields").
<b>RESERVED</b>	Returns information about the storage space occupied by the files (and file generations, since GEN=YES is implied) selected by the current FSTATUS command. Simultaneous specification of STANDARD, CATALOG, PASSWORD or ALL is ignored.
<b>SHARE</b>	Selects files and file generation groups on the basis of their shareability. If "\$userid." is a foreign user ID, SHARE=YES is automatically implied.
<b>=YES</b>	Returns information about files which are shareable.
<b>=NO</b>	Selects files which may be accessed only by their owners.
<b>=SPECIAL</b>	Selects files which the maintenance user ID \$SERVICE may access.



- SIZE** Requests information about files and file generations on the basis of the size of the reserved storage space.
- =FREESIZE** Selects files for which the number of reserved PAM pages is equal to the number of free pages.
  - =number** Selects files with precisely the specified number of reserved PAM pages.
  - =(number[,])** Selects files with at least the specified number of reserved PAM pages.
  - =(,number)** Selects files with not more than the specified number of reserved PAM pages.
  - =(number1,number2)** Selects files whose number of reserved PAM pages lies within the specified range (number1 < number2).
- SORT** Specifies how the catalog entries and/or path names are to be sorted for output.
- =FILENAM** The catalog entries and/or path names are sorted alphabetically for output; default value.
  - =NO** The catalog entries and/or path names are output in the order in which they are kept in the catalog.
- STANDARD** Returns information primarily about current storage space occupancy (see "Meaning of output fields", page 284).

The following catalog field entries are output for each selected file:

Output field	Meaning
FCBTYPE	Access method at file creation time
VSNTYPE	Volume type (public/private)
LASTPG	Number of occupied PAM pages
2ND ALLO	Secondary allocation

This is followed by total lines indicating the storage space occupied by the listed files, classified by volume types (public/private/tape). File generations are included only if GEN=YES is specified.

Output field	Meaning
:catid:	Catalog ID
FILES	Number of files per volume type
RES	Total number of reserved PAM pages
FREE	Total number of unused PAM pages
REL	Total number of PAM pages which can be released

- STATE** Selects files and file generations on the basis of whether they are open or whether a pseudo-close has been issued for them.
- =NOCLOS Returns information about output files which have been opened via an OPEN macro (OPEN OUTIN, INOUT or OUTPUT) and files which were not closed in a previous system run or because of job abortion. GEN=YES is implied.
  - =PCLOSE Selects files for which a pseudo-close has been issued (only for spoolout files).
- STORAGE LEVEL** Selects files depending on which storage level they are on (for more information see the "HSMS" manual). If more than one operand value is specified, they are logically ORed.
- =S0 Selects files on storage level S0.
  - =S1 Selects files migrated to storage level S1.
  - =S2 Selects files migrated to storage level S2.
- SUPPORT** Selects files, file generations and file generation groups on the basis of the type of volume on which they are stored. If a list of volume types is specified, these are logically ORed by the system.
- =PUBLIC Returns information about files, etc. stored on public volumes.
  - =PRDISC Returns information about files, etc. stored on private disks.
  - =TAPE Returns information about files/FGs/FGGs stored on tape.
- TYPE=FGG** Suppresses information about files and returns only information about file generation groups and file generations (see GEN=YES).


## TRAITS

Returns information primarily about the file and file generation attributes defined via the FILE command or FCB macro and about the specific attributes of file generation groups (see "Meaning of output fields", page 284).

The following catalog entry fields are output for each selected file:

Output field	Meaning
BASE	Appears for FGGs only: base value for relative generation numbers
BLKCTRL	Block control information
BLKTYPE	Block type (standard/non-standard block)
BLKSIZE	Block size (= buffer length)
BUFOFF	Buffer offset (tape files)
CODE	Code used (tape files)
DISP	Appears for FGGs only: action when the maximum number of simultaneous generations is reached
EXTCNT	Appears for disk files only: total number of extents for the file
FIRSTGN	Appears for FGGs only: oldest cataloged generation
FSEQ	Appears for tape files only: file sequence number
GEN	Appears for FGGs only: maximum number of concurrently cataloged generations
KEYLEN	ISAM key length
KEYPOS	ISAM index position
LABEL	Appears for tape files only: standard version of the labels
LASTGN	Appears for FGGs only: youngest/latest cataloged generation
LOGLEN	Length of the ISAM logical flag
RECFORM	Record format
RECSIZE	Record length
STORAGE-LEVEL	Storage hierarchy level
VALLEN	Length of the ISAM value flag
VALPROP	Propagation of the ISAM value flag
VSN/	Volume serial number of the volume
DEV/	Device type for the volume
EXT	Number of extents on the volume

This is followed by total lines indicating the storage space occupied by the listed files, classified by volume types (public/private/tape). File generations are included only if GEN=YES is specified. (see the description of "STANDARD").

- VOLUME=vsn Returns information on all files and file generation groups which have the specified volume serial number ("vsn") in their volume list.
- VTOC Specifies whether the requested information is to be taken from the VTOC (= Volume Table Of Contents) of a private disk or from the system file catalog \$TSOSCAT.
-  The VTOC operand is ignored for files that are not on a private volume, for partially qualified file names, or if it is specified in conjunction with GEN=YES.
- =NO Returns the current entry from the system catalog TSOSCAT; default value.
- =YES Returns the VTOC catalog entries (from the F1 label of a private disk) in accordance with the last current status in the entire computer network. The related volume must be assigned.
- The VTOC entry from the private volume replaces the corresponding TSOSCAT entry. This makes it possible to re-establish consistency between the VTOC and TSOSCAT entries if, for example, files of a shared private disk have been modified by a job running on another computer. If the specified file no longer exists on the private volume shown in the \$TSOSCAT entry, the catalog entry is deleted.
- All other keyword operands are ignored when VTOC=YES.

### Meaning of output fields

1. Header line

- n – numeric character
- a – alphanumeric character

Output field	Structure	Meaning
File size	nnnnnnn	7-digit number; specifies the number of PAM pages reserved for the file
	:catid:	Catalog ID of the pubset on which the file is cataloged
	\$userid.	User ID (max. 8 characters) of the file
	filename	Name of the permanent or temporary file, file generation or file generation group for which information is output

## 2. File attributes

Output field	Values	Description
ACCESS	WRITE	Write access to the file is permitted.
	READ	Only read access to the file is permitted, not write access.
ACCESS#	nnn	3-digit number indicating the number of accesses to the file. This is set to 0 when the file is created and incremented by 1 each time the file is opened. The maximum value is 255 and is not changed once this value has been reached.
ACL	YES/NO	Specifies whether or not the file is protected by an ACL entry.
AUDIT	NONE	No monitoring by system exit routines is defined for the file.
	ALL	All DMS actions for the files are monitored.
	SUCC	All successful DMS actions are monitored.
	FAIL	All unsuccessful DMS actions are monitored.
BACKUP	A	The file is saved in each backup run.
	B	The file is saved in backup runs for files with the attribute BACKUP=B/C/D.
	C	The file is saved in backup runs for files with the attribute BACKUP=C/D.
	D	The file is saved in backup runs for files with the attribute BACKUP=D.
	E	This file is not automatically saved by ARCHIVE.
BASE	nnnnn	Only for file generation groups: 5-digit number indicating the absolute generation number currently used as the base for relative generation numbers.

Output field	Values	Description
BLKCNT	nnnnnnn	For tape files: 7-digit number indicating the number of data blocks in the file.
BLKCTRL	DATA	The block control information is in the data block.
	NO	No block control information available.
	NONE	The file has not yet been opened.
	PAMKEY	Block control information is in the PAM key.
	UNKNOWN	The file has an undefined BLKCTRL value.
BLKSIZE	nnnnnn	6-digit number indicating the buffer size in bytes.
BLKTYPE	STD	Standard blocks: buffer size = 1 PAM page
	(STD,n)	The blocking factor ( $n \leq 16$ ) defined for the file.
BUFOFF	nn	For tape files: buffer offset.
CODE	EBCDIC/ISO7/ OWN	For tape files: code table with which the file was created.
CRDATE	yy-mm-dd	yy = year; mm = month; dd = day; creation date of the file, i.e. the date of the last write (update) access.
	00-00-00	The file has never been opened or has been erased with /ERASE file; DATA.
DESTROY	NO	No data destruction.
	YES	Data destruction for disk files; for tape files: residual data is overwritten when tapes are swapped and tape files are closed.
DEV	aaaaa	The device type specification for the volume (cf. the device type list in the appendix).

Output field	Values	Description
DISP	CYCLE	Only for file generation groups: when the GEN limit is reached, the oldest file and its catalog entry are erased.
	REUSE	Only for file generation groups: when the GEN limit is reached, the oldest file and its catalog entry are erased; for files on private volumes, the new generation is stored on the volume which becomes free.
	DELETE	Only for file generation groups: when the GEN limit is reached, all existing generations are erased.
	KEEP	Only for file generation groups: when the GEN limit is reached, no file generations are automatically erased; this is not done until a CATALOG command with the operand STATE=U is entered for the FGG.
EXDATE	yy-mm-dd	yy = year; mm = month; dd = day; The date until which the file is locked against write access, i.e. the file cannot be updated or erased.
EXPASS	NONE	No execute password is defined for the file.
	YES	The file is protected by an execute password, i.e. the file cannot be executed (/EXEC, /DO, /CALL, /ENTER) unless this password is specified.
EXT	nnn	The number of extents for the file on the volume designated by "VSN".
	*	The file has no extents on the volume designated by "VSN".
EXTCNT	nnn	Total number of extents for the file

Output field	Values	Description
FCBTYPE	NONE	There is only a catalog entry for the file; the file has never been opened; storage space may have been allocated (see CRDATE).
	ISAM/BTAM/ SAM/PAM/ PAM(PLAM)	The access method with which the file was created.
FIRSTGN	nnnnn	Only for file generation groups: 5-digit absolute generation number of the oldest generation cataloged for this file generation group.
FSEQ	nnnnn	For tape files: 5-digit number indicating the position of the file in an MF set.
GEN	nnnnn	Only for file generation groups: 5-digit number indicating the maximum number of generations which may be cataloged simultaneously for this file generation group.
GROUP	r-w-x	Group members' access rights
KEYLEN	nnn	Only for ISAM files: 3-digit number indicating the length of the ISAM key in the ISAM index (in bytes).
KEYPOS	nnnnn	Only for ISAM files: 5-digit number indicating the position of the ISAM key in the record.
LABEL	(STD,n)	For tape files with standard labels: the interchange level for DIN 66029, and thus the labels with which the file was created.
	NSTD	Identifies tape files with non-standard labels.
	NO	Identifies tape files without labels.
LADATE	yy-mm-dd	yy = year; mm = month; dd = day; the date of the last access to the file.



Output field	Values	Description
LARGE	NO	Save runs with ARCHIVE always save the entire file.
	YES	Save runs with ARCHIVE save only the PAM pages of the file which have been updated since the last save run.
LASTGN	nnnnn	Only for file generation groups: 5-digit absolute generation number of the youngest generation cataloged for the file generation group.
LASTPG	nnnnnnn	7-digit number indicating the number of PAM pages occupied by the file (only for files with standard blocks).
LOGLEN	nnn	Only for ISAM files which have a logical flag in the ISAM index: 3-digit number indicating the length of the logical flag in the ISAM index (in bytes).
MIGRATE	ALLOWED	HSMS is allowed to migrate the file to background storage.
	INHIBIT	The file must not be migrated.
RDPASS	NONE	No read password exists for the file.
	YES	A read password has been defined for the file, i.e. read access is possible only after specifying this password.
RECFORM		Specifies: a) the record type defined for the file, b) valid printer control characters
	(V,) (F,) (U,)	- the file contains variable-length records; - the file contains fixed-length records; - the file contains records with undefined length.
	(,A) (,M) (,N)	- ASA control characters are valid for the file; - EBCDIC control characters are valid for the file; - the file contains no printer control characters.

Output field	Values	Description
RECSIZE	nnnnn	5-digit number indicating the length in bytes defined for the records of the file (for RECFORM=F) or the maximum permissible record length (for RECFORM=V).
	00000	For files with FCBTYPE ≠ NONE and CRDATE ≠ NONE: together with RECFORM=V/U: maximum record length = BLKSIZE
SHARE	NO	The file is not shareable.
	YES	The file is shareable.
	SPECIAL	The file may be accessed under the user ID for maintenance personnel (\$SERVICE).
STORAGE-LEVEL	S1 / S2	For migrated files only (HSMS): the file has been migrated to the specified storage level.
OTHERS	r-w-x	Access rights for others
OWNER	r-w-x	Owner's access rights
VALLEN	nnn	Only for ISAM files which have a value flag in the ISAM index: 3-digit number indicating the length of the value flag in the ISAM index (in bytes).
VALPROP	MIN	Only for ISAM files which have a value flag in the ISAM index: the lowest value flag in each data or index block is placed in the related index entry in the next higher level.
	MAX	Only for ISAM files which have a value flag in the ISAM index: the highest value flag in each data or index block is placed in the related index entry in the next higher level.
VERSION	nnn	3-digit version number of the file.

Output field	Values	Description
VSN	PUBxaa	Catalog ID (x) and VSN (aa) or the public volume on which space is reserved for the file. Format for one-character catalog ID.
	xx.aaa xxx.aa xxxx.a	Catalog ID (x..) and VSN (a..) of the public volume on which space is reserved for the file. Format for a multi-character catalog ID
	aaaaaa	6-character alphanumeric volume serial number of the private volume on which the file occupies space.
VSNTYPE	PUB	The file is stored on public volumes (tape or disk).
	PVT	The file is stored on private volumes (tape or disk).
WRPASS	NONE	There is no write password for the file.
	YES	A write password is defined for the file, i.e. write access is possible only if this password is specified.
2ND ALLO	nnnnn	5-digit number indicating the value defined for secondary allocation.

## 3. Total lines

Output field	Values	Description
:catid:	catid	Catalog ID of the pubset on which the files are cataloged.
FREE	n	Number of reserved but unused PAM pages on the volume.
PRIVATE	n FILES	Number of files cataloged in the specified pubset which are stored on private volumes.
PUBLIC	n FILES	Number of files cataloged in this pubset.
REL	n	Number of PAM pages on the volume that could be released.
RES	n	Number of PAM blocks reserved on the volume.
SUM		Separate total lines for PUBLIC and PRIVATE; see above for a description of the output fields.
TAPE	n-FILES	Number of cataloged files stored on tape.

## Examples

*Example 1:* Standard function of the FSTATUS command

```
(IN)      FSTAT _____ (01)
(OUT)     0000030 :W:$US123456.EXAMP.FS.1
           0000000 :W:$US123456.EXAMPLES
           0000030 :W:$US123456.EXAMPLES.COPY
           0000003 :W:$US123456.MAX.DO.ASS.MAC
           0000003 :W:$US123456.MAX.DO.ASS.MAL
           0000030 :W:$US123456.MAX.PROTO.FSTAT.ALL.FGG
           0000030 :W:$US123456.MAX.PROTO.SEC.WAIT
           0000000 :W:$US123456.GROUP.LISTS (FGG)
           0000000 :W:$US123456.GROUP.PROTO (FGG)
           0000000 :W:$US123456.TEST
           0000000 :W:$US123456.TEXTST
           0000000 :W:$US123456.TEXT
           0000030 :W:$US123456.ZE.EXAMP.FSTAT.10
           0000030 :W:$US123456.ZE.EXAMP.FSTAT.11
           0000030 :W:$US123456.ZE.EXAMP.FSTAT.7
           0000030 :W:$US123456.ZE.EXAMP.FSTAT.8
           0000030 :W:$US123456.ZE.EXAMP.FSTAT.9
           :W: PUBLIC: 17 FILES. RES= 256, FREE= 215, REL= 201 PAGES
```

- (01) The FSTAT command without operands outputs an alphabetical list of all files cataloged for the user ID of the current job. The first column (e.g. 0000030) shows the number of PAM pages reserved for each file. This is followed by the path name: :catid:\$userid.filename.
- The last line shows the number of file names listed (FILES), the number of PAM pages reserved for these files (RES), how many of these pages are free (FREE), and how many could be released (REL). "PUBLIC" or "PRIVATE" indicates the volume type and :catid: shows the catalog ID under which the files are cataloged.

*Example 2: SORT=NO operand*

```
(IN)      FSTAT , SORT=NO _____ (01)
(OUT)     0000000 :W:$US123456.GROUP.LISTS (FGG)
          0000030 :W:$US123456.MAX.PROTO.FSTAT.ALL.FGG
          0000000 :W:$US123456.GROUP.PROTO (FGG)
          0000030 :W:$US123456.MAX.PROTO.SEC.WAIT
          0000000 :W:$US123456.EXAMPLES
          0000030 :W:$US123456.EXAMP.FS.1
          0000030 :W:$US123456.EXAMPLES.COPY
          0000030 :W:$US123456.ZE.EXAMP.FSTAT.7
          0000030 :W:$US123456.ZE.EXAMP.FSTAT.8
          0000000 :W:$US123456.TEXT
          0000030 :W:$US123456.ZE.EXAMP.FSTAT.9
          0000000 :W:$US123456.TEST
          0000000 :W:$US123456.TEXST
          0000030 :W:$US123456.ZE.EXAMP.FSTAT.10
          0000030 :W:$US123456.ZE.EXAMP.FSTAT.11
          :W: PUBLIC: 17 FILES. RES= 256, FREE= 214, REL= 201 PAGES
```

- (01) All files cataloged under this user ID are listed in the order in which they appear in the catalog.

*Example 3: Wildcards in the file name (cf. example 6c)*

```
(IN)      FSTAT *BEISP* _____ (01)
(OUT)     0000030 :W:$US123456.EXAMP.FS.1
           0000000 :W:$US123456.EXAMPLES
           0000030 :W:$US123456.EXAMPLES.COPY
           0000003 :W:$US123456.MAX.FSTAT.EXAMP.2
           0000030 :W:$US123456.ZE.EXAMP.FSTAT.2
           0000030 :W:$US123456.ZE.EXAMP.FSTAT.3
           :W: PUBLIC:      6 FILES. RES=      123, FREE=      115, REL=      111 PAGES

(IN)      FSTAT BEISP* _____ (02)
(OUT)     0000030 :W:$US123456.EXAMP.FS.1
           0000000 :W:$US123456.EXAMPLES
           0000030 :W:$US123456.EXAMPLES.COPY
           :W: PUBLIC:      3 FILES. RES=      60, FREE=      56, REL=      54 PAGES

(IN)      FSTAT EXAMP. _____ (03)
(OUT)     0000030 :W:$US123456.EXAMP.FS.1
           :W: PUBLIC:      1 FILE. RES=      30, FREE=      28, REL=      27 PAGES

(IN)      FSTAT EXAMPLES. _____ (04)
(OUT)     0000030 :W:$US123456.EXAMPLES.COPY
           :W: PUBLIC:      1 FILE. RES=      30, FREE=      28, REL=      27 PAGES

(IN)      FSTAT EXAMPLES _____ (05)
(OUT)     0000000 :W:$US123456.EXAMPLES
           :W: PUBLIC:      1 FILE. RES=      0, FREE=      0, REL=      0 PAGES

(IN)      FSTAT TE*T _____ (06)
(OUT)     0000000 :W:$US123456.TEST
           0000000 :W:$US123456.TEXST
           0000000 :W:$US123456.TEXT
           :W: PUBLIC:      3 FILES. RES=      0, FREE=      0, REL=      0 PAGES

(IN)      FSTAT TE/T _____ (07)
(OUT)     0000000 :W:$US123456.TEST
           0000000 :W:$US123456.TEXT
           :W: PUBLIC:      2 FILES. RES=      0, FREE=      0, REL=      0 PAGES
```

- (01) The character "\*" replaces any character string - in this case before and after the string "EXAMP"; i.e. all files which contain the string "EXAMP" anywhere in their names will be listed.
- (02) The character "\*" replaces any character string - in this case after the string "EXAMP": all files whose names begin with "EXAMP" will be listed.
- (03) The partial qualification "EXAMP." selects all files where the first partial name in the file name is "EXAMP."
- (04) This selects all files whose first partial name is "EXAMPLES."
- (05) "EXAMPLES" selects one, and only one file: the file with the fully qualified file name "EXAMPLES".
- (06) "\*" replaces any character string between the strings "TE" and "T".
- (07) The character "/" replaces just one character between the strings "TE" and "T".

Example 4: Partial qualification

```
(IN) FS MAX. _____ (01)
(OUT) 0000003 :W:$US123456.MAX.DO.ASS.MAC
      0000003 :W:$US123456.MAX.DO.ASS.MAL
      0000150 :W:$US123456.MAX.L.ANHANG
      0000030 :W:$US123456.MAX.L.EX1-57.MAC
      0000060 :W:$US123456.MAX.L.EX1-57.NEW
      0000030 :W:$US123456.MAX.L.EX60-92
      0000030 :W:$US123456.MAX.PROTO.FSTAT.FGG
      0000003 :W:$US123456.MAX.S.APPENDIX
      0000003 :W:$US123456.MAX.S.EX1-57
      0000003 :W:$US123456.MAX.S.EX1-57.NEW
      0000003 :W:$US123456.MAX.S.EX60-92
      :W: PUBLIC: 13 FILES. RES= 573, FREE= 115, REL= 96 PAGES
```

(01) All files which are addressed by the partial qualification "MAX." are selected.

Example 5: Catalog entry for a temporary file

```
(IN) FSTAT # _____ (01)
(OUT) % DMS0533 SPECIFIED FILE IN PUBSET 'W' NOT FOUND.
      CMD TERMINATED
(IN) CAT #TEMP _____ (02)
(IN) COPY MAX.FILE,#TEMP _____ (03)
(IN) FSTAT #TEMP,ALL _____ (04)
(OUT) 0000003 :W:$US123456.S.nnn.tsn.TEMP
      FCBTYPE = SAM VSNTYPE = PUB
      LASTPG = 00000001 2ND ALLO= 00003
      SHARE = NO ACCESS = WRITE
      ACL = NO AUDIT = NONE DESTROY = NO
      CRDATE = 1990-12-18 EXDATE = 1990-12-18 LADATE = 1990-12-18
      RDPASS = NONE WRPASS = NONE EXPASS = NONE
      ACCESS# = 001 VERSION = 001
      LARGE = NO BACKUP = E MIGRATE = INHIBIT
      BLKTYPE = STD BLKSIZE = 002048 BLKCTRL = PAMKEY
      RECFORM = (V,N) RECSIZE = 000000
      VSN/DEV/EXT = PUBW06 / D3480 / 001
      EXTCNT = 1
      :W: PUBLIC: 1 FILE RES= 3 FREE= 2 REL= 0 PAGES
```

- (01) The message shows that no temporary files exist for this job.
- (02) The CATALOG command creates a temporary file.
- (03) Temporary files can be processed just like permanent files: here, the file MAX.FILE is copied into the file #TEMP.
- (04) The catalog entry for a temporary file differs from that of a permanent file created with the default attributes in the field BACKUP=E. In addition, the internal path name of the file is shown: the prefix "#" is converted by the system into the character string S.nnn.tsn, where "nnn" is a unique number generated by the system and "tsn" is the TSN of the current job.



*Example 6a: CATALOG operand*

```
(IN) FSTAT MAX.FILE,CATALOG _____ (01)
(OUT) 0000003 :W:$US123456.MAX.FILE
      SHARE = NO          ACCESS = WRITE
      ACL = NO           AUDIT = NONE          DESTROY = NO
      CRDATE = 1990-12-18 EXDATE = 1990-12-18 LADATE = 1990-12-18
      RDPASS = NONE       WRPASS = NONE       EXPASS = NONE
      ACCESS# = 002      VERSION = 001
      LARGE = NO         BACKUP = A
      :W: PUBLIC:      1 FILE RES=      3 FREE=      2 REL=      0 PAGES
```

- (01) An extract from the catalog entry for MAX.FILE is returned. This contains the file attributes to be defined with the CATALOG command.

*Example 6b: RESERVED operand*

```
(IN) FSTAT ,R _____ (01)
(OUT) :W: PUBLIC:      64 FILES. RES=      780, FREE=      503, REL=      441 PAGES
      :W: PRIVATE:     3 FILES. RES=      48, FREE=      46, REL=      42 PAGES
```

- (01) Only the total lines for all files cataloged under the user ID of the current job in the default catalog (:W:) are shown.

*Example 6c: PASSWORD operand*

```
(IN) FSTAT TE*T*,PASSWORD _____ (01)
(OUT) 0000000 :W:$US123456.TEST
      RDPASS = YES       WRPASS = NONE       EXPASS = NONE
0000000 :W:$US123456.TEST.FILE
      RDPASS = NONE     WRPASS = YES        EXPASS = NONE
0000003 :W:$US123456.TEXTST
      RDPASS = NONE     WRPASS = NONE       EXPASS = NONE
0000003 :W:$US123456.TEXT
      RDPASS = NONE     WRPASS = NONE       EXPASS = NONE
0000003 :W:$US123456.TEXT.1
      RDPASS = NONE     WRPASS = NONE       EXPASS = NONE
0000003 :W:$US123456.TEXT.2
      RDPASS = NONE     WRPASS = NONE       EXPASS = NONE
      :W: PUBLIC:      6 FILES. RES=      12, FREE=      8, REL=      0 PAGES
```

- (01) The output shows for which files passwords have been defined.

*Example 7: Selection of files with specific attributes*

```
(IN)      FSTAT ,PASS=NONE ----- (01)
(OUT)    0000024 :W:$US123456.MAX.PROTO.COPY.2
          0000024*:W:$US123456.FILE.NEW
          0000000 :W:$US123456.GROUP.DATA (FGG)
          0000024*:W:$US123456.PRIV.DAT.1
          0000024*:W:$US123456.PRIV.DAT.2
          0000024*:W:$US123456.PRIV.DAT.3
          0000024*:W:$US123456.PRIV.DAT.4
          0000000 :W:$US123456.TEST
          0000024*:W:$US123456.TEST.PRIV
          0000000 :W:$US123456.TEST.PUB
          0000024*:W:$US123456.TEST.2.PRIV
          0000000 :W:$US123456.TEXST
          :W: PUBLIC:      5 FILES. RES=      24, FREE=      6, REL=     51 PAGES
          :W: PRIVATE:    7 FILES. RES=    168, FREE=    156, REL=    144 PAGES
```

**(01) All files which are not protected by passwords are listed.**

```
(IN)      FS ,STATE=NOCLOS ----- (01)
(OUT)    0000030 :W:$US123456.MAX.PROTO.FSTAT.ALL.FGG
          :W: PUBLIC:      1 FILE. RES=      30, FREE=      30, REL=     30 PAGES
```

**(01) All files open when the FSTATUS command is entered are listed.**

```
(IN)      FSTAT ,SUPPORT=PRDISC ----- (01)
(OUT)    0000024*:W:$US123456.FILE.NEW
          0000024*:W:$US123456.PRIV.DAT.1
          0000024*:W:$US123456.PRIV.DAT.2
          0000024*:W:$US123456.PRIV.DAT.3
          0000024*:W:$US123456.PRIV.DAT.4
          0000024*:W:$US123456.TEST.PRIV
          0000024*:W:$US123456.TEST.2.PRIV
          :W: PRIVATE:    7 FILES. RES=    168, FREE=    156, REL=    144 PAGES
```

**(01) All files stored on private disks are listed.**

```
(IN)      FSTAT ,SIZE=(2,8) ----- (01)
(OUT)    0000003 :W:$US123456.MAX.DO.ASS.MAC
          0000003 :W:$US123456.MAX.DO.ASS.MAL
          0000006 :W:$US123456.GS.D.B
          0000003 :W:$US123456.GS.DO.ASS
          0000003 :W:$US123456.GS.DO.EXAMP
          0000003 :W:$US123456.TEXT.DAT1
          0000003 :W:$US123456.TEXT.1
          0000003 :W:$US123456.TEXT.2
          :W: PUBLIC:      8 FILES. RES=      27, FREE=     10, REL=      0 PAGES
```

**(01) All files are listed for which at least two and not more than eight PAM pages are reserved.**

*Example 8: File generation groups and file generations**Example 8a: List of cataloged FGGs with their generations*

```
(IN)      FSTAT ,TYPE=FGG,GEN=YES _____ (01)
(OUT)     0000000 :W:$US123456.GROUP.LISTEN (FGG)
           0000003 :W:$US123456.GROUP.LISTEN(*0010)
           0000003 :W:$US123456.GROUP.LISTEN(*0011)
           0000000 :W:$US123456.GROUP.LISTEN(*0012)
           0000000 :W:$US123456.GROUP.PROTO (FGG)
           0000003 :W:$US123456.GROUP.PROTO (*0001)
           0000003 :W:$US123456.GROUP.PROTO (*0002)
           0000003 :W:$US123456.GROUP.PROTO (*0003)
           0000003 :W:$US123456.GROUP.PROTO (*0004)
           :W: PUBLIC:      9 FILES. RES=      21, FREE=      7, REL=      0 PAGES
```

- (01) The group entries of all cataloged file generation groups and the path names of all related and cataloged file generations are listed.

*Example 8b: Complete catalog entries of an FGG on public volume*

```
(IN)      FSTATUS MAX.GROUP.RES,GEN=YES,ALL _____ (01)
(OUT)     00000000 :N:$D89116.MAX.GROUP.RES (FGG)
           SHARE   = NO                ACCESS   = WRITE
           ACL      = NO                AUDIT    = NONE                DESTROY = NO
           CRDATE  = 1991-01-14        EXDATE   = 1991-01-14        LADATE  = NONE
           RDPASS  = NONE                WRPASS   = NONE                EXPASS  = NONE
           ACCESS# = 000                VERSION  = 000
           LARGE   = NO                BACKUP   = A                MIGRATE = ALLOWED
           GEN     = 00003              BASE     = 00002              DISP    = CYCLE
           FIRSTGN = 00003              LASTGN  = 00005

           00000003 :N:$D89116.MAX.GROUP.RES(*0003)
           FCBTYP  = SAM                VSNTYPE  = PUB
           LASTPG  = 00000001           2ND ALLO= 00003
           SHARE   = NO                ACCESS   = WRITE
           ACL      = NO                AUDIT    = NONE                DESTROY = NO
           CRDATE  = 1991-01-14        EXDATE   = 1991-01-14        LADATE  = 1991-01-14
           RDPASS  = NONE                WRPASS   = NONE                EXPASS  = NONE
           ACCESS# = 001                VERSION  = 001
           LARGE   = NO                BACKUP   = A                MIGRATE = ALLOWED
           BLKTYPE = STD                BLKSIZE  = 002048             BLKCTRL = PAMKEY
           RECFORM = (V,N)              RECSIZE  = 000000
           VSN/DEV/EXT = PUBN01 / D3480 / 001
           EXTCNT  = 1
```

```

00000003 :N:$D89116.MAX.GROUP.RES(*0004)
FCBTYPE = SAM                VSNTYPE = PUB
LASTPG  = 00000001          2ND ALLO= 00003
SHARE   = NO                ACCESS  = WRITE
ACL     = NO                AUDIT  = NONE
CRDATE  = 1991-01-14        EXDATE = 1991-01-14    DESTROY = NO
RDPASS  = NONE              WRPASS  = NONE                LADATE  = 1991-01-14
ACCESS# = 001               VERSION = 001                EXPASS  = NONE
LARGE   = NO                BACKUP  = A                MIGRATE = ALLOWED
BLKTYPE = STD               BLKSIZE = 002048          BLKCTRL = PAMKEY
RECFORM = (V,N)            RECSIZE = 000000
(OUT)   VSN/DEV/EXT =      PUBN02 / D3480 / 001
EXTCNT  = 1

00000003 :N:$D89116.MAX.GROUP.RES(*0005)
FCBTYPE = SAM                VSNTYPE = PUB
LASTPG  = 00000001          2ND ALLO= 00003
SHARE   = NO                ACCESS  = WRITE
ACL     = NO                AUDIT  = NONE
CRDATE  = 1991-01-14        EXDATE = 1991-01-14    DESTROY = NO
RDPASS  = NONE              WRPASS  = NONE                LADATE  = 1991-01-14
ACCESS# = 001               VERSION = 001                EXPASS  = NONE
LARGE   = NO                BACKUP  = A                MIGRATE = ALLOWED
BLKTYPE = STD               BLKSIZE = 002048          BLKCTRL = PAMKEY
RECFORM = (V,N)            RECSIZE = 000000
VSN/DEV/EXT =      PUBN03 / D3480 / 001
EXTCNT  = 1

:N:      PUBLIC:      4  FILES RES=      9  FREE=      6  REL=      0  PAGES

```

- (01) The complete catalog entries of the file generation group, the group entry and the file generations are to be output.

Example 8c: Complete catalog entries of an FGG on private disk

```

(IN)      FSTAT MAX.GROUP,GEN=Y,A,VOLUME=WORK01 _____ (01)
(OUT)    0000000*:W:$US123456.MAX.GROUP (FGG)
          SHARE = NO          ACCESS = WRITE
          ACL = NO            AUDIT = NONE          DESTROY = NO
          CRDATE = 1991-01-14 EXDATE = 1991-01-14 LADATE = NONE
          RDPASS = NONE       WRPASS = NONE       EXPASS = NONE
          ACCESS# = 000       VERSION = 000
          LARGE = NO         BACKUP = A           MIGRATE = ALLOWED
          GEN = 00005        BASE = 00000        DISP = CYCLE
          FIRSTGN = 00002    LASTGN = 00006
          VSN/DEV = WORK01 / D3480

0000003*:W:$US123456.MAX.GROUP(*0002)
          FCBTYP = NONE       VSNTYPE = PVT
          LASTPG = 00000000    2ND ALLO= 00003
          SHARE = NO          ACCESS = WRITE
          ACL = NO            AUDIT = NONE          DESTROY = NO
          CRDATE = NONE       EXDATE = NONE       LADATE = NONE
          RDPASS = NONE       WRPASS = NONE       EXPASS = NONE
          ACCESS# = 000       VERSION = 000
          LARGE = NO         BACKUP = A           MIGRATE = ALLOWED
          BLKTYPE = STD       BLKSIZE = 000000    BLKCTRL = NONE
          RECFORM = NONE     RECSIZE = 000000
          VSN/DEV/EXT = WORK01 / D3480 / 001
          EXTCNT = 1

0000003*:W:$US123456.MAX.GROUP(*0003)
          FCBTYP = NONE       VSNTYPE = PVT
          LASTPG = 00000000    2ND ALLO= 00003
          SHARE = NO          ACCESS = WRITE
          ACL = NO            AUDIT = NONE          DESTROY = NO
          CRDATE = NONE       EXDATE = NONE       LADATE = NONE
          RDPASS = NONE       WRPASS = NONE       EXPASS = NONE
          ACCESS# = 000       VERSION = 000
          LARGE = NO         BACKUP = A           MIGRATE = ALLOWED
          BLKTYPE = STD       BLKSIZE = 000000    BLKCTRL = NONE
          RECFORM = NONE     RECSIZE = 000000
          VSN/DEV/EXT = WORK01 / D3480 / 001
          EXTCNT = 1

```

```

0000003*:W:$US123456.MAX.GROUP(*0006)
FCBTYPE = NONE          VSNTYPE = PVT
LASTPG  = 00000000      2ND ALLO= 00003
SHARE   = NO            ACCESS  = WRITE
ACL     = NO            AUDIT   = NONE
CRDATE  = NONE          EXDATE  = NONE
RDPASS  = NONE          WRPASS  = NONE
ACCESS# = 000           VERSION = 000
LARGE   = NO            BACKUP  = A
BLKTYPE = STD           BLKSIZE = 000000
RECFORM = NONE          RECSIZE = 000000
VSN/DEV/EXT = WORK01 / D3480 / 001
EXTCNT  = 1
:W: PRIVATE: 4 FILES. RES= 9, FREE= 9, REL= 0 PAGES
    
```

- (01) The complete catalog entries for the file generation group (group entry) and for the cataloged file generations are listed. Note that the group entry is on private disk WORK01, but other private disks also contain generations (\*0004 and \*0005) belonging to this file generation group.

*Example 9: Catalog entry of a tape file*

```

(IN) FSTAT TAPEFILE,A
(OUT) :W:$US123456.TAPEFILE
FCBTYPE = SAM          VSNTYPE = PVT
BLKCNT  = 00000001
SHARE   = YES          ACCESS  = WRITE
ACL     = NO            AUDIT   = NONE
CRDATE  = 91-02-02     EXDATE  = 91-02-02
RDPASS  = NONE          WRPASS  = NONE
ACCESS# = 001           VERSION = 001
LARGE   = NO            BACKUP  = A
BLKTYPE = STD           BLKSIZE = 002048
RECFORM = (V,N)         RECSIZE = 002048
CODE    = EBCDIC        LABEL   = (STD,1)
BUFOFF  = 04
VSN/DEV = ( D5193A / TAPE )
:W: TAPE: 1 FILE.
    
```

## GETJV Output job variable value

Application group: Job variable functions (page 36)

The GETJV command is only available with the JV software product (see also "Job Variables" manual [11]).

### Command description

The GETJV command outputs the value of a user job variable or a special job variable to SYSOUT.

### Format and operand description

Operation	Operands
GETJV	$\left\{ \begin{array}{l} \text{jvid} \\ (\text{jvid}[, \text{start}] [, \text{length}]) \end{array} \right\}$ $[ , \left\{ \begin{array}{l} \text{CHAR} \\ \text{HEX} \end{array} \right\} ] [, \text{PASS}=\text{password}]$

jvid	This entry may contain one of the following types of job variable names:						
	<table> <tbody> <tr> <td>jvname</td> <td>Denotes the fully-qualified name of a permanent or temporary job variable.</td> </tr> <tr> <td>*jvlink</td> <td>Specifies a job variable link name.</td> </tr> <tr> <td>special-jvname</td> <td>Denotes the name of a special job variable.</td> </tr> </tbody> </table>	jvname	Denotes the fully-qualified name of a permanent or temporary job variable.	*jvlink	Specifies a job variable link name.	special-jvname	Denotes the name of a special job variable.
jvname	Denotes the fully-qualified name of a permanent or temporary job variable.						
*jvlink	Specifies a job variable link name.						
special-jvname	Denotes the name of a special job variable.						
start	Specifies the start position within the job variable value. If "start" is omitted, "1" is used. The start position must be between 1 and 256.						
length	Specifies the number of bytes to be output. $1 \leq \text{length} \leq 256$ . "start" plus "length" must not exceed 257. Default value: length of the job variable value.						



If the length of the job variable value, from the specified start position to the end of the JV value, differs from the number of bytes specified in "length", the smaller of these two values will be used. An attempt to read a non-existent (sub)string (e.g. a string whose start position lies outside the job variable value) will result in an error message.

<u>CHAR</u>	Specifies that the value is to be output as a character constant; default value.
HEX	Specifies that the value is to be output as a hexadecimal constant.
PASS=password	Specifies the password required for reading the job variable (see the CATVJ command). If the password is incorrect, the command is rejected, unless the correct password was previously entered in the password table by means of a PASSWORD command.

Examples: see the "Job Variables" manual [11].



## GETUS Read user switches

Application group: Job control (page 22 ff.)

### Command description

The GETUS command enables users to obtain information about the setting of their own and any other user ID's switches. The numbers of all switches set to ON are displayed.

32 user switches are provided for each user ID. They are numbered from 0 through 31 and stored in the JOIN file of the home pubset.

When a user ID is created, all of its 32 switches are set to OFF. Any switches that ARE subsequently set to ON will retain this setting even after LOGOFF.

The current job can be controlled by means of these user switches (SKIPUS or WHEN command).

### Format and operand description

Operation	Operands
GETUS	[userid]

userid                      User ID. Default value: user ID as specified with the LOGON command.

### Example

```
(IN)  LOGON XY,ABR00007,C'MAI'
      .....

(IN)  GETUS _____ (01)
(OUT) % SWITCHES ON EQUAL-
(OUT) % 1, 2, 3, 10
(IN)  GETUS BBMP _____ (02)
(OUT) % ALL SWITCHES SET OFF
```

(01) User switches 1, 2, 3 and 10 of user ID "XY" have already been set to on during previous jobs.

(02) All user switches for user ID "BBMP" are set to OFF.

For a further example, see the SKIPUS command.

## HELP Display message text

Application group: Job control (page 22 ff.)

### Command description

The HELP command displays the text of a system message on SYSOUT. You can also request descriptive text and specify the preferred language for the display of message text.

### Format and operand description

Operation	Operands
HELP	$\left[ \begin{array}{l} \{ \text{msgid} \} \\ \{ \underline{\text{LIST}} \} \end{array} \right]$ $[ , \text{INFORMATION} = \left[ \begin{array}{l} \{ \text{F} \} \\ \{ \text{C} \} \end{array} \right] \left[ \begin{array}{l} \{ \text{D} \} \\ \{ \text{M} \} \end{array} \right] ] ]$ $[ , \text{LANGUAGE} = \text{language} ]$

**msgid** 4- or 7-character message code. The corresponding message text is output to SYSOUT.

#### INFORMATION

**≡F** Displays the system message in normal form. F, together with D, is the default value. (F stands for "FULL MESSAGE").

**≡D** Displays the system message together with an explanation of the message text. D, together with F, is the default value. (D stands for "DESCRIPTION").

**=C** Displays the system message in coded abbreviated form (C stands for "CODE").

**=M** Displays only the message text. (M stands for "MINI").

#### LANGUAGE

Specifies the language for the output of system messages and descriptive text.

**=language** 1 letter, e.g. "E" for English, "D" for German. The symbols for additional languages are given in the "System Messages" manual [24] or may be obtained from the system administrator.

LIST

Repeats the system message which was last displayed for the task. Messages requested using the HELP command are not included here.

**Examples**

```
(IN)      FSTATUS XXXXX _____(01)
(OUT)     % DMS0533 REQUESTED FILE NOT CATALOGED ON PVS W. COMMAND TERMINATED

(IN)      HELP LAN=D _____(02)
(OUT)     % DMS0533 SPECIFIED FILE IN PVS W NOT FOUND. CMD TERMINATED
(OUT)     % This message is issued by DMS commands. The requested file is not
(OUT)     % cataloged in the requested PVS.
(OUT)     % None.

(IN)      HELP INF=C _____(03)
(OUT)     % DMS0533 W
(OUT)     % This message is issued by DMS commands. The requested file is not
(OUT)     % cataloged in the requested PVS.
(OUT)     % None.

(IN)      HELP INF=M _____(04)
(OUT)     % DMS0533 REQUESTED FILE NOT CATALOGED ON PVS W. CMD TERMINATED

(IN)      HELP INF=M,LAN=D _____(05)
(OUT)     % DMS0533 SPECIFIED FILE IN PVS W NOT FOUND. CMD TERMINATED

(IN)      HELP BLS0500,INF=M,LAN=D _____(06)
(OUT)     % BLS0500 PROGRAMM '(&00)', VERSION '(&01)' OF '(&02)' LOADED.
(OUT)     % COPYRIGHT (&04) (&05)

(IN)      HELP INF=M,LAN=D _____(07)
(OUT)     % DMS0533 SPECIFIED FILE IN PVS W NOT FOUND. CMD TERMINATED
```

- (01) The file specified by /FSTATUS is not cataloged; message DMS0533 is displayed.
- (02) The last message issued is to be repeated; the message text, together with descriptive text, is output in German (default).
- (03) The last message issued is displayed in coded abbreviated form.
- (04) The last message issued is displayed in "mini" format.
- (05) The last message issued is displayed in German and in "mini" format.
- (06) Message BLS0500 issued by the linking loader is requested by means of the HELP command. It is displayed in German and in "mini" format.
- (07) The text of the last message issued (DMS0533) is requested again. Message BLS0500 is ignored since it was requested explicitly using the HELP command.

## HOLD Place TFT entry in hold status

Application group: Device and volume reservation (page 29)

### Command description

The HOLD command is used to place an entry in the task file table (TFT) in the hold status. This serves to defer the subsequent RELEASE command (or REL macro) for this TFT entry until a DROP command with the relevant file link name is issued.

If the TFT entry is changed by means of the CHANGE command, it remains in hold status under its new name.

TFT entries in the hold status can be edited at any time by means of the FILE command.

### Format and operand description

Operation	Operands
HOLD	[link]

link Specifies the file link name of a TFT entry to be placed in the hold status. If no entry with this name existed previously, a new TFT entry is created for this link name. Further entries can then be made in this TFT entry by means of the FILE command.  
If "link" is not specified, the first TFT entry with link name C'\_\_\_\_\_' is processed.

**Examples:***Example 1:*

```

/FILE LINK=X, ...
/HOLD X
/RELEASE X
.
.
/DROP X

```

An entry with the name X is created in the task file table.

The hold status indicator is set for TFT entry X.

The entry X is not deleted, since it is subject to the HOLD indicator. However, the command (or REL macro) sets the RELEASE indicator in TFT entry X.

Only now does the RELEASE command or REL macro take effect, the TFT entry X being deleted and any associated private devices released.

*Example 2:*

```

/HOLD Y
/FILE LINK=Y, ...
.
.
/DROP Y
.
/RELEASE Y

```

An entry with the name Y is created in the TFT and its HOLD status indicator is set.

This command uses the existing TFT entry Y, and makes entries in it.

The HOLD status for entry Y is cancelled. As no RELEASE has been issued, no operation is performed.

Entry Y is no longer subject to HOLD. The RELEASE command thus has the immediate effect of deleting entry Y.

*Example 3:*

```

/FILE LINK=A
/HOLD A
/CHANGE A,B
/DROP A
/RELEASE B
/DROP B

```

The name of TFT entry A is changed to B, but everything else in the entry remains the same, e.g. the entry B is now subject to the HOLD status.

This command generates an error message as there is no longer an entry with this link name.

The RELEASE indicator is set in TFT entry B, but the command is deferred since HOLD is in force.

RELEASE takes effect, i.e. entry B (formerly entry Y) is deleted.

## IMPORT      Create catalog entry for private files

Application group: File processing (page 26 ff.)

### Command description

The IMPORT command catalogs files stored on private disks if they were created under the user ID of the job issuing the command. DMS takes the file attributes from the F1 label of the private disk and places them in the catalog entry. The command can process partially qualified file names, which means that the user can import several files with a single command.

When importing file generation groups with generations stored on different disks, please note that generations are cataloged only if the group entry already exists in the system catalog or is kept on the first disk to be imported. Otherwise, the catalog entries for the generations imported before the group entry will be missing. These generations must then be cataloged afterwards using an IMPORT or FILE command (STATE=FOREIGN).

The functions of the commands IMPORT and ERASE (operands CATALOG/DELETE-OR-EXPORT and VOLUME, respectively) are not exact opposites. When a volume is exported, DMS deletes the catalog entries for all files which occupy storage space on this volume. If the same volume is reimported, DMS creates catalog entries only for those files which begin on the volume (i.e. files which received space on the volume during their primary allocation).

Locked entries can be imported from the F1 label. If, however, entries in the user catalog have to be replaced (REPLACE=YES/ABS), the entries must not be locked and write access must be permitted.

## Format and operand description

Operation	Operands
IMPORT	<pre>[pathname, ]VOLUME=vsn, DEVICE=device  [, REPLACE={   YES   ABS   NO }]  [, LIST={   {     YES     NO     ONLY   }   (     {       YES       NO       ONLY     },     {       SYSLST       SYSOUT       BOTH     }   ) }]  [, GEN={   YES   NO }] [, PVSID=catid]</pre>

**pathname** stands for: [\$userid.]filename

Designates the files, file generation groups or file generations to be cataloged. If "pathname" is not specified, DMS catalogs all files, etc. which are stored under the user ID of the current job on the private disk specified in the VOLUME operand.



For file generations and file generation groups, create the group entry before cataloging the generations.

**userid** User ID; default value: the user ID specified in the LOGON command.

**filename** Fully or partially qualified name of a file, file generation group or file generation.

**DEVICE=device**

Device type for the volume; see the device table in appendix A.1 for possible entries for "device" (under the "Device type" column).

GEN	For file generation groups: specifies whether only the group entry or also the file generations stored on the same private disk is/are to be cataloged.
= <u>YES</u>	If the group entry is on the private disk, DMS catalogs the FGG and all related generations which start on this disk. If there is no group entry on the disk or in the user catalog, no file generations are cataloged; GEN=YES is the default value.
=NO	DMS transfers only the group entry of the FGG to the catalog.
LIST	Specifies how command execution is to be logged (see the table "Return codes/messages" below).
= <u>YES</u>	Command processing is logged; LIST=YES is the default value.
=NO	No information about command processing is returned.
= <u>ONLY</u>	Causes execution of the IMPORT command to be simulated, not actually carried out, i.e. the users receives a SYSOUT/SYSLST log which shows how the IMPORT command would have been executed. The log contains a list (depending on "pathname") of the files on the volume specified by VOLUME, together with the messages and information returned by the IMPORT command.  DMS does not check, at this time, whether file locks or protection attributes would prevent import of the files. For the actual import, you must ensure that the files are not locked and that write access is permitted.
=( <u>...,SYSLST</u> )	Command processing is logged to SYSLST. The log contains the return code and the message texts (see table "Return codes/messages" below); LIST=(...,SYSLST) is the default value.
=( <u>...,SYSOUT</u> )	Command execution is logged on SYSOUT. Only the return code is shown (see below).
=( <u>...,BOTH</u> )	Command execution is logged on SYSLST and SYSOUT. The SYSLST log contains the return code and the message texts, while the SYSOUT log contains only the return code.
PVSID=catid	Specifies the pubset in which the files are to be cataloged. If this is omitted, the catalog entries are created under the default catalog ID of the user ID.



---

REPLACE	Specifies whether an existing "old" catalog entry is to be overwritten.
= <u>NO</u>	DMS does not overwrite the existing catalog entry; REPLACE = NO is the default value.
=YES	The old catalog entry is overwritten if it does not match the specifications in the IMPORT command. <ul style="list-style-type: none"><li>– The cataloged file is stored on a public disk: the catalog entry is overwritten, which means that the public file is erased (providing the protection attributes permit this and the file is not locked; otherwise, the old catalog entry is left unchanged).</li><li>– The cataloged file is on private disk, but begins on a volume other than the one specified in the VOLUME operand: the catalog entry is overwritten (providing the protection attributes permit this and the file is not locked; otherwise, the old entry remains unchanged).</li></ul>
=ABS	The old catalog entry is overwritten even if the catalog entry and the specifications in the IMPORT command match. The return code shows whether the entry was overwritten (return code 8) or whether a file lock prevented overwriting (return code 9).
VOLUME=vs <sub>n</sub>	Volume serial number ("vs <sub>n</sub> ") of the private disk on which the files to be imported are stored.

## Return codes/messages

Code on SYSOUT	Message on SYSLST	Meaning
0	FILE DID NOT EXIST	There was no file with the same name and a new catalog entry has been created.
1	FILE HAS BEEN ERASED	A file of the same name already existed and has been overwritten; together with LIST=ONLY: a file with this name already existed, protection attributes have not been checked.
2	FILE EXISTS / REPLACE=NO	A file of the same name existed but was not overwritten; the REPLACE operand had the value NO.
3	FILE IS PROTECTED (ERASE ERROR OR FILE IS IN USE)	A file of the same name existed and could not be erased because of active protection functions (ACCESS=READ, WRPASS, etc.) or the file is locked because it is being processed.
4	ERROR ON CATALOG ACCESS	System error during catalog access.
5	FILE ALREADY ON PRIVATE	The file is already cataloged and is stored on the private disk specified by VOLUME.
6	ERROR ON VTOC ACCESS	System error during access to the F1 label of the private disk.
7	GENERATION OUT OF RANGE	Invalid attempts to import a file generation: the absolute generation number of the generation to be imported conflicts with the limits defined in the group entry.
8	C.E. HAS BEEN REPLACED	A catalog entry already existed for the specified disk and has been replaced.
9	C.E. IS PROTECTED (ERASE ERROR OR FILE IS IN USE)	A catalog entry already exists for the specified disk, but the file is locked.
A	INVALID FILENAME	The path name of the file to be imported (with pvsid and user ID) exceeds 54 characters.

**Examples**

```
(IN)      IMPORT , VOLUME=WORK02, DEVICE=D3480, LIST=(YES, SYSOUT)
(OUT)     0 :W:$US123456.FILE.NEW
          0 :W:$US123456.TEST.PRIV
          0 :W:$US123456.TEST.2.PRIV
```

All files stored for \$US123456 (user ID of the current job) on private disk WORK02 are imported.

Return code on SYSOUT: 0 (a new catalog entry was created and the file imported).

```
(IN)      ERASE FILE.NEW
(IN)      IMPORT FILE.NEW, VOLUME=WORK02, DEVICE=D3480, LIST=(YES, BOTH)
(OUT)     % DMS05FE REQUESTED FILE(S) NOT FOUND
```

The "private" file FILE.NEW and its catalog entry were completely erased.

```
(IN)      ERASE TEST.PRIV, DEL-OR-EXP
(IN)      IMPORT TEST.PRIV, VOLUME=WORK02, DEVICE=D3480, LIST=(YES, BOTH)
(OUT)     0 :W:$US123456.TEST.PRIV
```

```
A * * *   I M P O R T   O U T P U T   L I S T   * * *           10:24:42  11/04/87
/IMPORT TEST.PRIV, VOLUME=WORK02, DEVICE=D3480, LIST=(YES, BOTH)
      OLD USER-ID CODE                NEW USER-ID CODE                FILE NAME
                                      FILE DID NOT EXIST                :W:$US123456.TEST.PRIV
```

Since only the catalog entry for TEST.PRIV was deleted, the file can be imported again with the IMPORT command.

Return code on SYSOUT: 0

Message on SYSLST: FILE DID NOT EXIST. (The IMPORT OUTPUT LIST is written to SYSLST).

**REPLACE operand**

```
(IN)      ERASE TEST.PRIV, DEL-OR-EXP
(IN)      COPY TEST.2.PRIV, TEST.PRIV
(IN)      FSTAT TEST.PRIV
(OUT)     0000024 :W:$US123456.TEST.PRIV
          :W: PUBLIC:      1 FILE. RES=      24, FREE=      22, REL=      21 PAGES
```

The catalog entry for the "private" file TEST.PRIV is deleted and a new file TEST.PRIV is created on pubset W. 24 PAM pages are reserved for this file, but it actually occupies only 2 PAM pages (FREE = 22 PAGES).

# IMPORT

---

```
(IN)      IMPORT TEST.PRIV,VOLUME=WORK02,DEVICE=D3480,LIST=(YES,BOTH)
(OUT)     2 :W:$US123456.TEST.PRIV
```

```
A * * *   I M P O R T   O U T P U T   L I S T   * * *           10:26:01  11/04/87
/IMPORT TEST.PRIV,VOLUME=WORK02,DEVICE=D3480,LIST=(YES,BOTH)
      OLD USER-ID CODE                NEW USER-ID CODE                FILE NAME
                                      FILE EXISTS/REPLACE=NO          :W:$US123456.TEST.PRIV
```

The IMPORT command is to catalog the "private" file TEST.PRIV again. Since the default value is REPLACE=NO, the pubset file TEST.PRIV is not erased.

Return code on SYSOUT: 2

Message on SYSLST: FILE EXISTS/REPLACE=NO. (The IMPORT OUTPUT LIST is written to SYSLST.)

```
(IN)      IMPORT TEST.PRIV,VOLUME=WORK02,DEVICE=D3480,LIST=(YES,BOTH),REPLACE=YES
(OUT)     1 :W:$US123456.TEST.PRIV
```

```
A * * *   I M P O R T   O U T P U T   L I S T   * * *           10:26:45  11/04/87
/IMPORT TEST.PRIV,VOLUME=WORK02,DEVICE=D3480,LIST=(YES,BOTH),REPLACE=YES
      OLD USER-ID CODE                NEW USER-ID CODE                FILE NAME
                                      FILE HAS BEEN ERASED           :W:$US123456.TEST.PRIV
```

```
(IN)      FSTAT TEST.PRIV
(OUT)     0000024*:W:$US123456.TEST.PRIV
          :W: PRIVATE: 1 FILE. RES= 24, FREE= 20, REL= 18 PAGES
```

This time, REPLACE=YES is specified in the IMPORT command. As can be seen from the messages, the pubset file TEST.PRIV is erased and replaced by the "private" file TEST.PRIV.

Return code on SYSOUT: 1

Message on SYSLST: FILE HAS BEEN ERASED. (The IMPORT OUTPUT LIST is written to SYSLST.)

The output from the FSTATUS command shows that TEST.PRIV is stored on private disk ("\*" before the path name) and that the new file occupies 2 PAM pages more than the old one (FREE = 20 PAGES).

## File generation groups

```
(IN)          IMPORT GROUP.PRIV,DEVICE=D3480,VOLUME=WORK02

A * * *      I M P O R T   O U T P U T   L I S T   * * *          10:35:35  11/04/87
/IMPORT GROUP.PRIV,DEVICE=D3480,VOLUME=WORK02
  OLD USER-ID CODE          NEW USER-ID CODE          FILE NAME
  FILE DID NOT EXIST        :W:$US123456.GROUP.PRIV
  FILE DID NOT EXIST        :W:$US123456.GROUP.PRIV(*0005)
  FILE DID NOT EXIST        :W:$US123456.GROUP.PRIV(*0006)
  FILE DID NOT EXIST        :W:$US123456.GROUP.PRIV(*0007)
  FILE DID NOT EXIST        :W:$US123456.GROUP.PRIV(*0008)
  FILE DID NOT EXIST        :W:$US123456.GROUP.PRIV(*0009)
```

```
(IN)          FSTAT GROUP.PRIV,GEN=YES
(OUT)         0000000*:W:$US123456.GROUP.PRIV (FGG)
              0000003*:W:$US123456.GROUP.PRIV(*0005)
              0000003*:W:$US123456.GROUP.PRIV(*0006)
              0000003*:W:$US123456.GROUP.PRIV(*0007)
              0000003*:W:$US123456.GROUP.PRIV(*0008)
              0000003*:W:$US123456.GROUP.PRIV(*0009)
              :W: PRIVATE:      6 FILES. RES=      15, FREE=      15, REL=      0 PAGES
```

The file generation group GROUP.PRIV stored on private disk WORK02 is imported with all of its generations (\*5 - \*9).

```
(IN)          FILE GROUP.PRIV(*10),DEVICE=D3480,VOLUME=WORK01
(IN)          COPY ZE.EXAMP.ER.3,GROUP.PRIV(*10)
(IN)          FILE GROUP.PRIV(*11),DEVICE=D3480,VOLUME=WORK01
(IN)          FILE GROUP.PRIV(*12),DEVICE=D3480,VOLUME=WORK01
(IN)          FILE GROUP.PRIV(*13),DEVICE=D3480,VOLUME=WORK01
(IN)          COPY ZE.EXAMP.ER.3,GROUP.PRIV(*11)
(IN)          COPY ZE.EXAMP.ER.3,GROUP.PRIV(*12)
(IN)          COPY ZE.EXAMP.ER.3,GROUP.PRIV(*13)
(IN)          FSTAT GROUP.PRIV,GEN=YES
(OUT)         0000000*:W:$US123456.GROUP.PRIV (FGG)
              0000003*:W:$US123456.GROUP.PRIV(*0009)
              0000024*:W:$US123456.GROUP.PRIV(*0010)
              0000024*:W:$US123456.GROUP.PRIV(*0011)
              0000024*:W:$US123456.GROUP.PRIV(*0012)
              0000024*:W:$US123456.GROUP.PRIV(*0013)
              :W: PRIVATE:      6 FILES. RES=      99, FREE=      95, REL=      84 PAGES
```

The new generations (\*10 to \*13) are created on private disk WORK01. The FSTATUS command shows that the file generation group now consists of the generations \*9 - \*13.

```

(IN)      FSTAT GROUP.PRIV,GEN=YES,ALL
(OUT)    0000000*:W:$US123456.GROUP.PRIV (FGG)
          SHARE = NO      ACCESS = WRITE
          ACCESS# = 000    CRDATE = 87-11-04  EXDATE = 87-11-04  LADATE = NONE
          RDPASS = NONE    WRPASS = NONE      EXPASS = NONE
          VERSION = 000    LARGE = NO      BACKUP = A
          DESTROY = NO     AUDIT = NONE
          GEN = 00005     BASE = 00000    LASTGN = 00013    FIRSTGN = 00009
          DISP = CYCLE
          VSN/DEV/EXT =    WORK02/D3480
0000003*:W:$US123456.GROUP.PRIV(*0009)
          FCBTYP = NONE    VSNTYPE = PVT      LASTPG = 0000000    2ND ALLO= 00009
          SHARE = NO      ACCESS = WRITE
          ACCESS# = 000    CRDATE = NONE      EXDATE = NONE      LADATE = NONE
          RDPASS = NONE    WRPASS = NONE      EXPASS = NONE
          VERSION = 000    LARGE = NO      BACKUP = A
          DESTROY = NO     AUDIT = NONE
          BLKTYPE = NONE   BLKSIZE = 000000    RECFORM = NONE     RECSIZE = 00000
          VSN/DEV/EXT =    WORK02/D3480/001
          EXTCNT = 1
0000024*:W:$US123456.GROUP.PRIV(*0010)
          FCBTYP = SAM     VSNTYPE = PVT      LASTPG = 0000001    2ND ALLO= 00009
          SHARE = NO      ACCESS = WRITE
          ACCESS# = 001    CRDATE = 87-11-04  EXDATE = 87-11-04  LADATE = 87-11-04
          RDPASS = NONE    WRPASS = NONE      EXPASS = NONE
          VERSION = 001    LARGE = NO      BACKUP = A
          DESTROY = NO     AUDIT = NONE
          BLKTYPE = STD    BLKSIZE = 002048    RECFORM = (V,M)    RECSIZE = 00000
          VSN/DEV/EXT =    WORK01/D3480/003
          EXTCNT = 3

```

- - - to save space, the catalog entries for generations \*11 to \*13 have been omitted here - - -

:W: PRIVATE: 6 FILES. RES= 99, FREE= 95, REL= 84 PAGES

A comparison of the catalog entries shows that the group entry is cataloged on disk WORK02, which also holds the oldest generation \*9. Generation \*10 (and generations \*11 to \*13) are on disk WORK01.

The entire file generation group is deleted from the file catalog.

```
(IN) ERASE ,DEL-OR-EXP,VOLUME=WORK02
(OUT) % DMS0516 DELETE ALL FILES WITH NOT FULLY QUALIFIED FILENAME
      ':W:$US123456.'? REPLY (Y=YES; N=NO)?
(IN) Y
(IN) ERASE ,DEL-OR-EXP,VOLUME=WORK01
(OUT) % DMS0516 DELETE ALL FILES WITH NOT FULLY QUALIFIED FILENAME
      ':W:$US123456.'? REPLY (Y=YES; N=NO)?
(IN) Y
```

On WORK01, a new group entry GROUP.PRIV with attributes which differ from those of the entry on WORK02 is created.

```
(IN) CAT GROUP.PRIV,GEN=3,BASE=12,FIRST=11,DEVICE=D3480,VOLUME=WORK01
(IN) FSTAT GROUP.PRIV,ALL
(OUT) 0000000*:W:$US123456.GROUP.PRIV (FGG)
      SHARE = NO ACCESS = WRITE
      ACCESS# = 000 CRDATE = 87-11-04 EXDATE = 87-11-04 LADATE = NONE
      RDPASS = NONE WRPASS = NONE EXPASS = NONE
      VERSION = 000 LARGE = NO BACKUP = A
      DESTROY = NO AUDIT = NONE
      GEN = 00003 BASE = 00012 LASTGN = 00012 FIRSTGN = 00011
      DISP = CYCLE
      VSN/DEV/EXT = WORK01/D3480
      :W: PRIVATE: 1 FILE. RES= 0, FREE= 0, REL= 0 PAGES
```

Comparison of the two group entries:

GROUP.PRIV (fields in group entry)		WORK01	WORK02
Max. number of generations	GEN=	3	5
Base value:	BASE=	12	0
Youngest generation:	LASTGN=	12	13
Oldest generation:	FIRSTGN=	11	9

With the following **IMPORT** command, **GROUP.PRIV** is to be imported in accordance with the group entry on disk **WORK01**.

```
(IN)      IMPORT GROUP.PRIV,VOLUME=WORK01,DEVICE=D3480,LIST=(YES,BOTH)
(OUT)     2 :W:$US123456.GROUP.PRIV
(OUT)     7 :W:$US123456.GROUP.PRIV(*0010)
(OUT)     0 :W:$US123456.GROUP.PRIV(*0011)
(OUT)     0 :W:$US123456.GROUP.PRIV(*0012)
(OUT)     7 :W:$US123456.GROUP.PRIV(*0013)
(IN)      FSTAT GROUP.PRIV,GEN=YES
(OUT)     0000000*:W:$US123456.GROUP.PRIV (FGG)
0000024*:W:$US123456.GROUP.PRIV(*0011)
0000024*:W:$US123456.GROUP.PRIV(*0012)
:W: PRIVATE: 3 FILES. RES= 48, FREE= 46, REL= 42 PAGES
```

As the messages and the **FSTATUS** command show, only the file generations which match the group entry (\*11 and \*12) are imported.

The following example shows the effect of an **IMPORT** command with **WORK02** specified in the **VOLUME** operand: **IMPORT** of **GROUP.PRIV** from **WORK02** after exporting the newly created **GROUP.PRIV** from **WORK01**.

```
(IN)      ER GROUP.PRIV,DEL-OR-EXP,VOLUME=WORK01
(IN)      IMPORT GROUP.PRIV,VOLUME=WORK02,DEVICE=D3480,LIST=(YES,BOTH)
(OUT)     0 :W:$US123456.GROUP.PRIV
(OUT)     0 :W:$US123456.GROUP.PRIV(*0009)
(IN)      FSTAT GROUP.PRIV,ALL
(OUT)     0000000*:W:$US123456.GROUP.PRIV (FGG)
          SHARE = NO      ACCESS = WRITE
          ACCESS# = 000    CRDATE = 87-11-04  EXDATE = 87-11-04  LADATE = NONE
          RDPASS = NONE    WRPASS = NONE      EXPASS = NONE
          VERSION = 000    LARGE = NO      BACKUP = A
          DESTROY = NO     AUDIT = NONE
          GEN = 00005     BASE = 00000    LASTGN = 00013    FIRSTGN = 00009
          DISP = CYCLE
          VSN/DEV/EXT = WORK02/D3480
:W: PRIVATE: 1 FILE. RES= 0, FREE= 0, REL= 0 PAGES
```

As the **IMPORT** messages show, the group entry and generation \*9 were imported again. The other (missing) generations are then imported with the following **IMPORT** command -- as shown by the **FSTATUS** command.



```
(IN)      IMPORT GROUP.PRIV,GEN=YES,VOLUME=WORK01,DEVICE=D3480,LIST=(YES,BOTH)
(OUT)     2 :W:$US123456.GROUP.PRIV
(OUT)     0 :W:$US123456.GROUP.PRIV(*0010)
(OUT)     0 :W:$US123456.GROUP.PRIV(*0011)
(OUT)     0 :W:$US123456.GROUP.PRIV(*0012)
(OUT)     0 :W:$US123456.GROUP.PRIV(*0013)
(IN)      FSTAT GROUP.PRIV,GEN=YES
(OUT)     0000000*:W:$US123456.GROUP.PRIV (FGG)
          0000003*:W:$US123456.GROUP.PRIV(*0009)
          0000024*:W:$US123456.GROUP.PRIV(*0010)
          0000024*:W:$US123456.GROUP.PRIV(*0011)
          0000024*:W:$US123456.GROUP.PRIV(*0012)
          0000024*:W:$US123456.GROUP.PRIV(*0013)
          :W: PRIVATE:    6 FILES. RES=    99, FREE=    95, REL=    84 PAGES
```

## INTR Start interrupt routine for loaded program

Application group: Program control (page 30)

### Command description

The INTR command effects continuation of an interrupted program run at the address defined in the STXIT macro for the event "communication with the program" (for details on use of STXIT, see the "Executive Macros" manual [5]). If the program does not include an STXIT routine for this "event", the INTR command is rejected with an error message.

### Format and operand description

Operation	Operands
INTR	[text]

text

Specifies a message which is moved to a buffer in the user program defined by a STXIT macro. If the latter does not specify a buffer, the message is ignored. If no text is supplied, but a buffer has been reserved, then only a null byte is transferred.

The maximum text length is 64 characters. Longer texts are truncated on the right, shorter ones are suffixed by a null byte to denote the end of the text in the user buffer.

Leading and trailing blanks (X'40') are suppressed. The buffer must begin on a word boundary.



Comments are not permissible in the INTR command as this would be transferred - together with the text specified - to the buffer of the user program.

**Example:**

For demonstration purposes, a loaded interactive program runs in a loop and consequently keeps issuing the same message via the data terminal:

```
INTERMEDIATE RESULT 25.
INTERMEDIATE RESULT 25.
INTERMEDIATE RESULT 25.
INTERMEDIATE RESULT 25.

/INTR _____ (01)
HERE STXIT EXIT FOR INTR INTERRUPT
.....
```

- (01) The program is interrupted by pressing the K2 function key (ESCAPE or BREAK) and control passes to system mode. The INTR command causes control to return to the loaded program, not to the break point but to the address of the STXIT routine which issues a message here.

If the system responds to an INTR command with the message:

```
%EXC0924 NO OPERATOR ROUTINE IN STXIT; COMMAND IGNORED
```

or

```
%EXC0923 STXIT NOT INITIALIZED; COMMAND IGNORED.
```

then the loaded program has no STXIT routine. In this case, you can make use of the RESUME command, e.g. commence at the start of the program with "RESUME L'0" (see the "Interactive Debugging Aid" manual [4]).

## **LOAD Load program**

Application group: program control (page 30)

### **Command description**

The LOAD command allows you to load programs to memory without initiating them. The dynamic linking loader (DLL) links object modules and linking load modules (LLM). The static loader ELDE loads the load modules. For more details, see the "BLS" manual [22] and "LMS" manual [14].

In order to initiate a loaded program, you must issue the RESUME command.

LOAD is rejected if the file with the linking/load module is protected by a read password which is not entered in the task password list (PASSWORD command).

The loading of programs cannot be nested. If a LOAD command is entered while a program is loaded, this program is unloaded, and the one named in the LOAD command is loaded instead. Although no error flag is set, any ABEND-STXIT routine defined in the unloaded program is activated, unless the LOAD command was called via the CMD macro (see the "Executive Macros" manual [5]).

The EXECUTE command can be used instead of LOAD + RESUME to load and then initiate a program.

**Formats and operand descriptions: See EXECUTE command (page 194 and following pages).**

## Examples

### Example 1

```

(IN)      LOAD U.LOAD _____ (01)
(OUT)    % BLS0517 MODULE TICK LOADED.
(IN)     STA P
(OUT)    NAME          TSN TYPE          C SIZE CURR-CMD PROGRAM-NAME
(OUT)    SALEM        8550 3 DIALOG    2     1 STA          :A:PA123456:U.LOAD
(IN)     RESUME
(OUT)    HERE IS TICK
...
(IN)     SYSFILE SYSDTA=(SYSCMD)
(IN)     EXEC $ASSEMB
(OUT)    % BLS0500 PROGRAM 'ASSEMB', VERSION '991' OF '87-01-15 LOADED.
(OUT)    V29.1X30 OF SIEMENS BS 2000 ASSEMBLER READY
(OUT)    GIVE ASSEMBLER OPTIONS !
(IN)     *COMOPT SOURCE=S1.SAL.PROG.TICK,ERRFIL,SAVLST,ISD
(OUT)    GIVE ASSEMBLER OPTIONS !
(IN)     *END HALT
(OUT)    FLAGS IN 00000 STATEMENTS, 000 PRIVILEGED FLAGS, 000 MNOTES
(OUT)    HIGHEST ERROR-WEIGHT : -

(IN)     LOAD *,IDA=YES,TIME=10 _____ (02)
(OUT)    % BLS0001 DLL VER 915
(OUT)    % BLS0517 MODULE TICK LOADED.
(IN)     AT FEHL
(IN)     RESUME
(OUT)    HERE IS TICK
(OUT)    INTERRUPTED AT 000014
(IN)     R

```

- (01) The load module in the file U.LOAD is loaded. However processing does not commence until a RESUME command is given. The memory requirements for this program are taken from the header record of the program. The permitted CPU time is the system default value.
- (02) In the language processor run, the source program in the S1.SAL.PROG.TICK file was compiled, and the generated object module with the name TICK was written to the temporary EAM object module file "\*". The LOAD command loads the program for a test run which is initially allocated 10 seconds of CPU time. The entry "IDA=YES" guarantees that symbolic addresses, as specified in the source program, can also be used in IDA commands.

## *Example 2*

If the LOAD command is used to load modules which require resident memory space, then the operand CLASSII must be specified. In interactive mode, you can employ the "/STATUS BIAS" command (CORE entry) to find out in advance whether the system has enough memory for the resident programs.

```
/LOAD LOADMOD.1, CLASSII=(6,4,6) _____ (01)
```

(01) The program LOADMOD.1 requires a total of 6 pages of virtual address space, 4 of which are resident.

## LOGOFF Terminate job

Application group: Job control (page 22 ff.)

### Command description

The LOGOFF command is used to terminate a job. The operating system then releases the virtual memory pages and devices used by that job and assigns the system output files to printer or tape.

For newly created file generations, information output to SYSOUT includes the name of the file generation groups concerned, the base value and the first and current generations.

If a LOGOFF command is specified while a program is loaded, any ABEND-STXIT routine that has been defined is activated. This does not apply if the LOGOFF command is called via the CMD macro (see the "Executive Macros" manual [5]).

Any monitoring job variable is set to \$T.

### Format and operand description

Operation	Operands
LOGOFF	$\left[ \begin{array}{l} \text{BUT} [ , \text{TAPE} ] \\ \text{TAPE} [ , \text{BUT} ] \\ \text{BUT} [ , \text{NOSPOOL} ] \\ \text{NOSPOOL} [ , \text{BUT} ] \end{array} \right]$

**BUT** This operand applies only to interactive users and is ignored in batch mode. It enables you to indicate that you wish to begin a new job when the current one is terminated and therefore do not want to clear down the existing connection to the terminal. If the BUT entry is omitted, the connection is cleared down.

**NOSPOOL** This operand suppresses output of the system files SYSLST and SYSOUT (for MSG=H in the LOGON or OPTION command) to the printer, and SYSOPT to the punch. The system files SYSLST and SYSOUT are not printed out if they are empty at the start of LOGOFF processing.

TAPE Causes the system files to be spooled out to tape rather than printer.



TAPE is only supported for reasons of compatibility. The PRINT/PUNCH command should be used for spooling out system files to tape.

## Examples

### Example 1

```
(IN)      .ABCDEF LOGON PA123456,M1234MON
(OUT)     % JMS066 JOB ABCDEF ACCEPTED ON 91-01-09 AT 16:10, TSN = 9KL0
(IN)      LOGOFF
(OUT)     % EXC0419 LOGOFF AT 1617 ON 91-01-09, FOR TSN 9KL0 _____ (01)
(OUT)     % EXC0421 USED CPU TIME : 1.5425 , SERVICE UNITS: 000003646 — (02)
```

- (01) The job with TSN 9KL0 is terminated at 16.17 on 9 January, 1991.
- (02) The amount of CPU time (1.5425 seconds) and number of service units (3646) used by the job are output.  
The connection to the terminal is cleared immediately after system message EXC0421 has been issued.

### Example 2

```
(IN)      LOGOFF BUT
(OUT)     % EXC0419 LOGOFF AT 1624 ON 85-07-29, FOR TSN 9MM4
(OUT)     % EXC0421 USED CPU TIME : 1.6914 , SERVICE UNITS: 000004433
(OUT)     % JMS0150 INSTALLATION ' H90-P', BS2000 VERSION 'V100', HOST
           'D123ZE01': PLEASE ENTER 'SET-LOGON-PARAMETERS' OR '?'
```

The connection to the terminal remains intact. A new LOGON command can be entered immediately.

### Example 3

```
(IN)      LOGOFF
(OUT)     GROUP: :V:$PA123456.GROUP1
           BASE: 0000      FIRST: 0001      CURRENT: 0002
(OUT)     % EXC0419 LOGOFF AT 1643 ON 85-07-29, FOR TSN 9922
(OUT)     % EXC0421 USED CPU TIME : 2.0780 , SERVICE UNITS: 000012345
```

Information on file generation group 1 relating to the base value, first, and current generations is displayed.



## LOGON      Initiate job

Application group: Job control (page 22 ff.)

### Command description

The LOGON command initiates any job, whether in interactive or batch mode.

The entries in the **LOGON** command identify the user (i.e. check his authorization to access the system), characterize the job and determine the manner in which job execution is logged.

The access authorization entries are checked against the entry in the JOIN file; furthermore, additional entries on job class and job attributes (job and run priority, system resources) are checked against the entry in the job class definition. These entries are available to you via the command **SHOW-USER-ATTRIBUTES** or **SHOW-JOB-CLASS**.

If the PRIORITY and NTL (no time limit) specifications in the JOIN file and the job class definition do not match, the value which is most favorable to you is selected.

The operands JOB-PRIO, RERUN, FLUSH, START and REPEAT are only evaluated in batch mode.

For an ENTER job the following should be noted: the operands in the LOGON command are ignored unless the (ENTER) file is started at the console.

If the LOGON command is repeatedly rejected in interactive mode, the connection can be cleared down with LOGOFF.




The PRIORITY and MSG operands are only supported for reasons of compatibility. Instead, the operand RUN-PRIO or the operand RUN-PRIO together with START=IMMEDIATELY (for PRIORITY=(p,EXPRESS)) and LOG should be used.

## Format and operand description

Name	Operation	Operands
[.jobname]	LOGON	<pre> userid,accountno [,password]  [,JOB-CLASS={   job-class   *STD }]  [,MONJV=jvname] [,JVPASS=password]  [,JOB-PRIO={   jprio   STD }]  [,RERUN={   NO   YES }]  [,FLUSH={   NO   YES }]  [,START={   STD   SOON   IMMEDIATELY   WITHIN ( {     HOURS=hour [,MINUTES=minute]     [HOURS=hour,]MINUTES=minute   } )   AT ( [DATE=yy-mm-dd,]TIME=hh:mm)   EARLIEST ( [DATE=yy-mm-dd,]TIME=hh:mm)   LATEST ( [DATE=yy-mm-dd,]TIME=hh:mm)   AT-STREAM-STARTUP }]  [,REPEAT={   STD   NO   DAILY   WEEKLY   PERIOD ( {     HOURS=hour [,MINUTES=minute]     [HOURS=hour,]MINUTES=minute   } )   AT-STREAM-STARTUP }] </pre>

Name	Operation	Operands
	LOGON (cont.)	<p>[ , RUN-PRIO = { rprio } ]</p> <p>[ , TIME = { t } ]</p> <p>[ , PRINT = { number } ]</p> <p>[ , PUNCH = { number } ]</p> <p>[ , LOG = ( LISTING = { NO } , HARDCOPY = { NO } ) ]</p> <p>[ , JOB-PAR = { *NO } ]</p> <p>[ , PRIORITY = { P } ]</p> <p>[ , MSG = { F } ] [L] [H] [T]</p>

jobname	Specifies a job name of 1 to 8 characters (for valid characters see page 7). This name (in addition to the user ID and account number) is printed in capitals on the job listings and the listing of the spooling jobs initiated by this job.
userid	User ID. The user ID is assigned by the system administrator. It identifies the user to the system.
accountno	Account number. The account number is assigned by the system administrator and indicates your account (system resource utilization).
password	Password protecting the user ID. Specified in the form of a C string (C'...') or X string (X'...'). You can define, modify or delete this password using the PSWORD command if you have the appropriate authorization in the JOIN entry. In batch jobs, the password is not logged to SYSOUT, i.e. it does not appear in the job's printer listings.

 Interactive password protection:  
If you have defined a password and do not specify it in the LOGON command, you will receive the message "... PLEASE ENTER PASSWORD", after which you can enter the password without it being displayed on the screen.

## FLUSH

=YES

Removes the job from the job queue if it was not processed by the end of the session (SHUTDOWN).

FLUSH is only evaluated in batch mode.




=NO





Job remains in the queue. (The next system session must begin with a warm or selective start.); default value.



Job control with RERUN/FLUSH:

- If FLUSH=YES and RERUN=YES were specified and the job was interrupted during the preceding session, FLUSH=NO is used in the next system run. This guarantees that the job remains in the job queue, even if it is not started in this session.
- A monitoring job variable is set to "\$S" when the job is repeated.



JOB-CLASS	Indicates a job class into which the job is to be integrated. Authorization to access the various job classes can be displayed by using the SHOW-USER-ATTRIBUTES or SHOW-JOB-CLASS command.
=jobclass	Name of the job class.
=*STD	Default value: (standard) job class predefined for the user or the system.
JOB-PAR	Enables specification of additional attributes for the selected job class, provided the system administrator has defined and announced them.
=*NO	No additional attributes; default value.
='string'	Any sequence of characters assigned by the system administrator to identify additional job class attributes.
JOB-PRIO	Determines the priority (relative to the other jobs) for starting a batch job. Otherwise, this entry has no effect on job processing. The operand is only evaluated in batch mode.
=jprio	Job priority, where $MAXIMUM \leq jprio \leq 9$ . The lower the value, the higher (greater) the job priority (urgency). The MAXIMUM value is defined in the job class definition.
=STD	Default value for the job class; preset value.
JVPASS	Indicates a password that authorizes access to the monitoring job variable. The password must be specified in an ENTER command if job distribution is requested; if job distribution is not requested, the password can also be issued via the PASSWORD command. JVPASS is ignored if MONJV is not specified.
=password	Password for the job variable "jvname".

- LOG=(...)** Indicates whether the job execution log is also to be output to SYSLST (LISTING=YES) and/or hardcopy printer (HARDCOPY=YES).  
Default: LISTING=NO; HARDCOPY=NO.
-  For output in LINE mode and HARDCOPY=YES, the entire message is output (same effect as the OHOM=YES operand in the WROUT and WRTRD macros).
- MONJV** Indicates a job variable which you can use to monitor a job. While the job is executing, the operating system assigns to the job variable the value "\$S" (job in job queue), \$R (job being processed), \$T (job terminated normally) or \$A (job terminated prematurely).
-  The MONJV operand is only available with the software product JV (see the "Job Variables" manual [11]).
- =jvname** Name of the job variable.
- MSG** Defines the type of logging for further job execution. The MSG operand is only supported for reasons of compatibility. MSG is ignored if LOG has been specified.
-  Please note for error messages: In some cases, the order of messages output does not match the order of their causes, as a result of the internal execution order.
- =F** System messages are output in full-length form to the system file SYSOUT (F for "Full Message").
- =C** System messages are output to SYSOUT in coded abbreviated form (C for "Code").
- =L** Console messages and operator responses for this job are logged to SYSOUT. Operator commands controlling job execution (e.g. priority changes) as well as general (system-related) warnings and error messages for the operator are not output here (L for "Log").
- If MSG=LH is entered, messages logged to SYSLST will also include the time of day at which they were issued.

- =H All messages to SYSOUT are also logged to SYSLST (H for "Hold Message").
- Exception: System messages requiring a response from the user, as well as the message "ABNORMAL PROGRAM TERMINATION".
-  In line mode (MODE=LINE), logging is performed line-by-line, i.e. NL control characters are analyzed. In format mode (MODE=FORM), logging takes place continuously, i.e. representation is not true to format; NL control characters are not analyzed.
- =T In interactive mode, messages to SYSOUT are also output to a hardcopy printer. Formatted messages as well as user entries are not printed.
-  It should be noted that, for output in LINE mode and MSG=T, the whole message is regarded as one output unit (same output as for OHOM=YES in the WROUT and WRTR macros).
- Denotes the maximum number of records output by the task (in summary form) to the system files SYSLST, SYSLST01, SYSLST02 through SYSLST99. Records written simultaneously to SYSOUT and SYSLST (LOG=LISTING or MSG=FH entry) do not count.
- =number Number of records, where  $0 \leq \text{number} \leq 999999$ .
- =NO Number of records is unlimited.
- =STD Default value of selected job class; preset value.
-  Exceeding the specified number:
- in batch mode, the task is terminated abnormally;
  - in interactive mode, you specify whether the task is to continue or terminate. If continued, output will again occur until "number" is reached.
- PRIORITY Determines the priority (relative to other tasks) for job execution.
-  PRIORITY is only supported for reasons of compatibility. Instead, the operand RUN-PRIO should be used, or RUN-PRIO in conjunction with START=IMMEDIATELY.

- =p** Run priority, where  $\text{MAXIMUM} \leq p \leq 255$ .  
The lower the value, the higher (greater) the priority (urgency).  
The **MAXIMUM** value is defined both in the job class definition and in the **JOIN** file, and can be displayed on the screen with the **SHOW-JOB-CLASS** or **SHOW-USER-ATTRIBUTES** command. If the values do not match, the limit value most advantageous to you is selected.  
Preset value: default value of selected job class.
-  This default value is also used if illegal values are entered for "p".
- =(p,EXPRESS)** The **EXPRESS** entry is only effective for batch jobs. It causes the batch job to start immediately.  
The entry has no further effect on job execution.
- PUNCH** Indicates the maximum number of records output by the task to the system file **SYSOPT**.
- =number** Number of records, where  $0 \leq \text{number} \leq 999999$ .
- =NO** Number of records is unlimited.
- =STD** Default value of the selected job class; preset value.
-  Exceeding the specified number:  
– in batch mode, the task is terminated abnormally;  
– in interactive mode, you specify whether the task is to continue or terminate. If continued, output will again occur until "number" is reached.
- REPEAT** Indicates a time interval after which the job is to be periodically started.  
The repetition is regarded as a job sequence.  
"J(0)" indicates the first job run, "J(1)" the first repetition, ... , "J(n)" the nth repetition. With the start of job J(i) the repetition J(i+1) is also created, where  $i \geq 0$ .
- REPEAT** is only evaluated in batch mode.
- =STD** Standard job class selection; default value.
- =NO** The job will not be repeated.



=DAILY	Daily repetition at the time specified in START.
=WEEKLY	Weekly repetition at the time specified in START.
=PERIOD(...)	Repetition at the specified interval (in hours and minutes), where $0 \leq \text{hour} \leq 23$ ; $0 \leq \text{minute} \leq 59$ .
=AT-STREAM-STARTUP	Repetition following each job scheduler startup.
	 The repeat values NO, DAILY, WEEKLY, PERIOD and AT-STREAM-STARTUP are only allowed if they are also allowed in the job class definition (see the SHOW-JOB-CLASS command).
RERUN	The job will be reinitiated in the next BS2000 session if execution was interrupted by an unrecoverable error or end of session.
=YES	RERUN is only evaluated in batch mode.
=NO	The job will not be reinitiated (default value).
RUN-PRIO	Determines the priority (relative to other tasks) for job execution.
=rprio	Run priority, where $\text{MAXIMUM} \leq \text{rprio} \leq 255$ . The lower the value, the higher (greater) the priority (urgency). The MAXIMUM value is defined both in the job class definition and in the JOIN file, and can be displayed with the SHOW-USER-ATTRIBUTES or SHOW-JOB-CLASS command. If these values do not match, the limit value most advantageous will be selected.
=STD	Standard job class selection; default value.
	 The default value is also used if the values entered for "rprio" are illegal.
START	Indicates a time (timespan) for starting the job.  Specifications for start time: DATE = yy-mm-dd : Date (yy=year, mm=month, dd=day). TIME = hh:mm : time-of-day (hh=hour, mm=Minute).  Hyphens and colons in DATE= and TIME= must be specified. Example: 31 May 1990 at 15.08 AT (DATE=90-05-31, TIME= 15:08).

The following applies to TIME:  $00 \leq hh \leq 23$ ;  $00 \leq mm \leq 59$ .

The ENTER command is rejected if the specified date or time is in the past. A year in the past specified as yy is interpreted as the year 20yy.



The start values SOON, IMMEDIATELY, WITHIN, AT, EARLIEST, LATEST and AT-STREAM-STARTUP are only permitted if they are also permitted in the job class definition (see the SHOW-JOB-CLASS command).

= <u>STD</u>	Standard job class selection; default value.
=SOON	The job is to be started as soon as possible, depending on its priority.
=IMMEDIATELY	The job is to start immediately.
=WITHIN(...)	The job is to start within the specified period (in hours and minutes). $0 \leq \text{hour} \leq 23$ ; $0 \leq \text{minute} \leq 59$ .
=AT(...)	The job is to start at exactly the specified time (date, time-of-day).
=EARLIEST(...)	The job is to start at or after the specified time (date, time-of-day).
=LATEST(...)	The job is to start at or before the specified time (date, time-of-day).
=AT-STREAM-STARTUP	The job is to start following job scheduler startup.
TIME	Indicates the maximum amount of CPU time (in seconds) which the task may use. The maximum amount which may be specified is determined by the selected job class. In batch mode, the task is terminated once the specified time has expired. In interactive mode, the task is not terminated; instead a warning is displayed indicating that the specified time has expired.
=t	CPU time in seconds, where $0 \leq t \leq \text{maximum CPU time}$ .
=NTL	No time limit. The task will run with unlimited CPU time.
= <u>STD</u>	Standard job class selection; default value.

## Combining the START and REPEAT operands

START	REPEAT		
	AT-STREAM-STARTUP	DAILY or WEEKLY	PERIOD
IMMEDIATELY or SOON	a)	c)	c)
AT or EARLIEST	a)	d)	f)
LATEST or WITHIN	a)	c)	g)
AT-STREAM- STARTUP	b)	e)	h)

- a) The first and all subsequent job starts take place as defined.
- b) The first job start takes place with START=AT-STREAM-STARTUP.  
All subsequent starts take place following job scheduler startup with START=SOON.
- c) Time base for repeat cycle is the time the job was accepted.
- d) The specified time (START=..., TIME=...) is the time base for the repeat cycle.
- e) The first job start takes place following job scheduler startup. This start time is the time base for the repeat cycle. Subsequent starts take place with START=SOON.
- f) The specified time (START=..., TIME=...) is the time base for the repeat cycle. The second and all subsequent starts take place with START=SOON.
- g) Time base for the repeat cycle is the time the job was accepted. All subsequent starts take place with START=SOON.
- h) Time base for the repeat cycle is the first start time. The first job start takes place following job scheduler startup. Subsequent starts take place with START=SOON.
- The nth repetition ( $n \geq 1$ ) of a job is not started until the (n-1)th execution has terminated.
  - Aborting the current job J(n) has no effect on the start of job J(n+1); ( $n \geq 0$ ).
  - Aborting the entire job: both the current job J(n) and the subsequent job J(n+1) must be aborted, where  $n \geq 0$  (CANCEL command or make job J(n) the last job of the repeat sequence with the command MODIFY-JOB tsn, REPEAT=NO).

**Examples:***Example 1*

```
(OUT)      % JMS0150 INSTALLATION ' H90-P ', BS2000 VERSION 'V100', HOST
           'D123ZE12': PLEASE ENTER '/SET-LOGON-PARAMETERS' OR '?'
(IN)       LOGON XYZ,ABR07MAN,C'DONALD' _____ (01)
(OUT)      % JMS0066 JOB ACCEPTED ON 91-01-31 AT 16:10, TSN = 4277 _____ (02)
```

- (01) After receiving the request "PLEASE ENTER...", the user enters the LOGON command with the user ID "XYZ" and the account number "ABR07MAN". His ID is protected by the password "DONALD".
- (02) The job runs under TSN 4277. LOGON was processed at 16.10 on 31 January, 1991.

*Example 2*

The following command is entered at a terminal:

```
/LOGON DDD,A0001234,C'PPWW',MSG=FB
```

Job execution is additionally logged to the system file SYSLST (see page 697 ff.). All outputs to the terminal are marked with "(OUT)" and all inputs at the terminal with "(IN)". For example, the response to the LOGON command is logged as follows:

```
(OUT)      % JMS0066 JOB ACCEPTED ON 85-07-31 AT 16:10, TSN = 4277
```

*Example 3*

A batch job is started with the following command:

```
/LOGON DAGOBERT,G9988ELD
```

The ID is not protected by a password. Therefore, the command is accepted.

The listing (to SYSOUT) for this batch job begins with:

```
EXC0238 JOB SCHEDULED AT TIME 1253 ON DATE=85-08-24 FOR TSN=4397
/LOGON DAGOBERT,G9988ELD
```

The information supplied by the user in the LOGON command of the ENTER file, though included in the printer listing, is ignored by the system.

*Example 4*

```
/LOGON ABC,ABT00014,RUN-PRIO=93,MSG=C,TIME=120
```

The batch job thus initiated is given run priority 93. All system messages are displayed in abbreviated form. The task may use up to 120 CPU seconds.

*Example 5*

```
/.GR1 LOGON HELDG,S8063050,RUN-PRIO=200.
```

The job thus initiated is given the string "GR1" as job name. The run priority is 200. It only affects subsequent job execution, not job initiation.

## **MODIFY-JOB      Modify job attributes**

Application group: job control (page 22 ff.)

### **Command description**

The MODIFY-JOB command enables you to change the job attributes (job data) of a batch job. In particular, it can be used to modify the entries made for the job in the LOGON or ENTER command regarding:

- the job class
- the job scheduling priority
- the specified start time
- job sequences (repeats)
- job repetition following abortion
- job parameters.

The entries regarding job class, job priority, start time and job parameters can only be changed if the job has not yet started (i.e. is still in the job queue).

The job must run under the user's own user ID (entry in the LOGON command).

The modified job data must be consistent with the (limit) values defined for the job class concerned (job class definition).

If operand errors occur, the command is completely rejected.


The start time of a repeat job (job sequence) cannot be modified.

When the job class is changed, the job receives the default values of the new job class, unless specific job attributes have been set using the MODIFY-JOB command.



**Format**

Operation	Operands
<pre> {MODIFY-JOB} {MOD-J}           </pre>	<pre> {TSN=tsn} {MONJV=jvname}  [,JOB-CLASS=job-class]  [,JOB-PRIO=jprio]  [,RERUN={   YES   NO }]  [,START={   STD   SOON   IMMEDIATELY   WITHIN({     HOURS=hour[,MINUTES=minute]     [HOURS=hour,]MINUTES=minute   })   AT([DATE=yy-mm-dd,]TIME=hh:mm)   EARLIEST([DATE=yy-mm-dd,]TIME=hh:mm)   LATEST([DATE=yy-mm-dd,]TIME=hh:mm)   AT-STREAM-STARTUP }]  [,REPEAT={   STD   NO   DAILY   WEEKLY   PERIOD({     HOURS=hour[,MINUTES=minute]     [HOURS=hour,]MINUTES=minute   })   AT-STREAM-STARTUP }]  [,JOB-PAR={   *NO   string }]           </pre>

**Operand description (in alphabetical order)**

JOB-CLASS	Indicates a job class into which the job is to be placed. Information on the authorization to access various job classes can be obtained with the SHOW-USER-ATTRIBUTES or SHOW-JOB-CLASS command.
=jobclass	Name of the job class.
=*STD	Default value: (standard) job class predefined for the user or the system.
JOB-PAR	Enables specification of additional attributes for the selected job class, provided the system administrator has defined and announced them.
=*NO	No additional attributes; preset value.
='string'	Any sequence of characters assigned by the system administrator to identify additional job class attributes.
JOB-PRIO	Determines the priority (relative to the other jobs) for starting a batch job. Otherwise, this entry has no effect on job processing.
=jprio	Job priority, where $MAXIMUM \leq jprio \leq 9$ . The lower the value, the higher (greater) the job priority (urgency). The MAXIMUM value is defined in the job class definition and can be displayed on the screen with the SHOW-JOB-CLASS command.
=*STD	The default value for the job class is used; preset value.
MONJV	Indicates a job variable with which you can monitor your job. During job execution, the operating system assigns the following values to the job variable: <ul style="list-style-type: none"> <li>– \$S (job in job queue),</li> <li>– \$R (job being processed),</li> <li>– \$T (job terminated normally) or</li> <li>– \$A (job terminated prematurely).</li> </ul> <div style="margin-left: 20px;">  <span style="display: inline-block; vertical-align: middle;">The MONJV is only available with the JV software product. (see also "Job Variables" manual [11]).</span> </div>
=jvname	Name of job variable



REPEAT	Indicates a time interval at which the job is to be periodically started. The repetition is regarded as a job sequence. "J(0)" indicates the first job run, "J(1)" the first repetition, ... , "J(n)" the nth repetition. With the start of job J(i) the repetition J(i+1) is also created, where $i \geq 0$ .
= <u>STD</u>	Default value of the selected job class; preset value.
=NO	The job will not be repeated.
=DAILY	Daily repetition at the time specified in START.
=WEEKLY	Weekly repetition at the time specified in START.
=PERIOD(...)	Repetition at the specified interval (in hours and minutes), where $0 \leq \text{hour} \leq 23$ ; $0 \leq \text{minute} \leq 59$ .
=AT-STREAM-STARTUP	Repetition following each job scheduler startup.
	 The repeat values NO, DAILY, WEEKLY, PERIOD and AT-STREAM-STARTUP are only allowed if they are also allowed in the job class definition (see the SHOW-JOB-CLASS command).
RERUN	
=YES	The job will be reinitiated in the next BS2000 session if execution was interrupted by a a serious system error or end-of-session.
	 A monitoring job variable is set to "\$S" when the job is repeated.
= <u>NO</u>	The job will not be reinitiated; default value.

START	<p>Indicates a start time (timespan) for the job.</p> <p>Specifications for the start time:          DATE = yy-mm-dd : date (yy=year, mm=month, dd=day).          TIME = hh:mm : time-of-day (hh=hour, mm=Minute).</p> <p>Hyphens and colons in DATE= and TIME= must be specified.          Example: 31 May 1990 at 15:08 AT (DATE=90-05-31, TIME=15:08).</p> <p>The following applies to TIME: <math>00 \leq hh \leq 23</math>; <math>00 \leq mm \leq 59</math>.</p> <p>The ENTER command is rejected if the specified month or time is in the past. A year in the past specified with "yy" is interpreted as the year 20yy.</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px 0;">!</div> <p>The start values SOON, IMMEDIATELY, WITHIN, AT, EARLIEST, LATEST and AT-STREAM-STARTUP are only permitted if they are also permitted in the job class definition (see the SHOW-JOB-CLASS command).</p>
=STD	Standard job class selection; default value.
=SOON	The job is to be started as soon as possible, depending on its priority.
=IMMEDIATELY	The job is to start immediately.
=WITHIN(...)	The job is to start within the specified period (in hours and minutes). $0 \leq \text{hour} \leq 23$ ; $0 \leq \text{minute} \leq 59$ .
=AT(...)	The job is to start exactly at the specified time (date, time of day).
=EARLIEST(...)	The job is to start at or after the specified time (date, time of day).
=LATEST(...)	The job is to start at or before the specified time (date, time of day).
=AT-STREAM-STARTUP	The job is to start following job scheduler startup.
TSN	Denotes the TSN (Task Sequence Number) of the job whose job data is to be modified.
=tsn	Task sequence number of the job

**Combining the START and REPEAT operands:**

START	REPEAT		
	AT-STREAM-STARTUP	DAILY or WEEKLY	PERIOD
IMMEDIATELY or SOON	a)	c)	c)
AT or EARLIEST	a)	d)	f)
LATEST or WITHIN	a)	c)	g)
AT-STREAM- STARTUP	b)	e)	h)

- a) The first and all subsequent job starts take place as defined.
- b) The first job start takes place with START=AT-STREAM-STARTUP.  
All subsequent starts take place following job scheduler startup with START=SOON.
- c) Time base for repeat cycle is the time the job was accepted.
- d) The specified time (START=..., TIME=...) is the time base for the repeat cycle.
- e) The first job start takes place following job scheduler startup. This start time is the time base for the repeat cycle. Subsequent starts take place with START=SOON.
- f) The specified time (START=..., TIME=...) is the time base for the repeat cycle. The second and all subsequent starts take place with START=SOON.
- g) Time base for the repeat cycle is the time the job was accepted. All subsequent starts take place with START=SOON.
- h) Time base for the repeat cycle is the first start time. The first job start takes place following job scheduler startup. Subsequent starts take place with START=SOON.
  - The nth repetition ( $n \geq 1$ ) of a job is not started until the (n-1)th has terminated.
  - Aborting the current job J(n) has no effect on the start of job J(n+1); ( $n \geq 0$ ).
  - Aborting the entire job: both the current job J(n) and the subsequent job J(n+1) must be aborted, where  $n \geq 0$  (CANCEL command or make job J(n) the last job of the repeat sequence with the command MODIFY-JOB tsn, REPEAT=NO.
  - The job class for the (repeat) job J(n) can only be changed after J(n-1) has terminated.
  - REPEAT=NO can only be specified for job J(n) after J(n-1) has terminated.

## MODIFY-JV-CONDITIONALLY      Check and set job variable

Application group: Job variable functions (page 36)

The MODIFY-JV-CONDITIONALLY command is only available with the JV software product (see also "Job Variables" manual [11]).

### Command description

This command enables you to read a job variable in a command procedure (DO, CALL or ENTER) and to compare its current value with a character string. If the job variable value is the same as the compare value, the job variable is set to the desired new value and command sequence resumes processing at a selectable point. If the job variable value is not the same as the compare value, processing continues with the next command.

While MODIFY-JV-CONDITIONALLY is executing, the job variable to be checked is protected from access by other jobs.

Any job variable specified under SET-VALUE is not protected from access by other jobs.

MODIFY-JV-CONDITIONALLY is rejected in interactive mode.

### Format and operand description

Operation	Operands
MODIFY-JV- CONDITIONALLY	JV-IDENTIFICATION= { JV-NAME (JV-NAME=jvname1 [, POSITION=start [, LENGTH=length]]) [ LINK-NAME (LINK-NAME=*jvlink1 [, POSITION=start [, LENGTH=length]]) ] } , IF-VALUE=compare-value , SET-VALUE={ newvalue { jvname2 } , LABEL=. flag { *jvlink2 } } [ , PASSWORD={ NONE { password } } ]

JV-IDENTIFICATION=(...)

Denotes the job variable to be checked and, if necessary, to be set. The JV name or the JV link name can be specified.

- `jrname1` Fully-qualified name of a permanent or temporary cataloged job variable.
- `*jrlink1` Valid JV link name.
- `start` First byte of the job variable value to be checked/set. "start" must be between 1 and 256. Default value: 1.
- `length` Number of bytes to be checked/set. The sum of "start" and "length" must not exceed the length of the job variable to be checked by more than 1.  
  
If the substring to be modified is smaller than the length of the current job variable value, the latter remains unchanged.  
If "length" is omitted, the implicit length of IF-VALUE or SET-VALUE is used.

IF-VALUE

=compare-value

"compare-value" is a character or hexadecimal constant in the form C' ... ' or X' ... ' which is compared with the value of the job variable to be checked. If they match, the job variable is set to the new value specified in SET-VALUE and processing continues from the flag defined with LABEL=. If they do not match, processing will continue with the next command that follows in the procedure.

SET-VALUE

=newvalue

Specifies a character or hexadecimal constant in the form C' ... ' or X' ... '. "jrname1" or "\*jrlink1" is set to this value if its previous value is identical to the constant specified in IF-VALUE.

=jrname2

Specifies the fully-qualified name of a cataloged permanent or temporary job variable. "jrname1" or "\*jrlink1" is set to this value if its previous value is identical to the constant specified in IF-VALUE.

=\*jrlink2

Specifies a valid job variable link name. For its meaning, see "jrname2".

### PASSWORD

=password

Specifies the password required for accessing the job variable to be checked or set (cf. CATJV).

This operand can be omitted if the password was previously entered using a PASSWORD command.

Default value: no password required (NONE).

= NONE

No password necessary; default value.

LABEL=.flag

Specifies the branch destination within the command sequence to which a branch will be made if the value of "jvname1" or "jvlink1" matches "compare-value" (see IF-VALUE).

See "Job Variables" manual for examples [11].

# MODIFY-MSG-ATTRIBUTES      Define language for message output

Application group: Job control (page 22 ff.)

## Command description

The **MODIFY-MSG-ATTRIBUTES** command enables you to define a particular language to be used for message output, as well as specifying the message files (system-wide or task-specific) to which this definition is to apply.

The default values for message output (scope of the message files, language for message output) are defined in the JOIN entry. The command SHOW-USER-ATTRIBUTES or SHOW-MSG-DEFAULTS can be used to interrogate these defaults and display them in the MES-SEARCH and MES-LANG fields. If no values are specified there, the values set at system generation time are used.

## Format and operand description

Operation	Operands
MODIFY-MSG-ATTRIBUTES	<pre> SEARCH= {   *UNCHANGED   *ALL   *TASK   *STD }  , TASK-LANGUAGE= {   *UNCHANGED   *STD   language   *NO } </pre>

**SEARCH**                      Restricts the search for messages to message texts in the language defined for this task (TASK-LAN=... operand). If there is no message text in the specified language for the message, it is output in the standard language as defined in the JOIN entry or at system generation time. The SEARCH= entries do not affect message output with the HELP command.

=\*UNCHANGED

The definitions made for the current task run continue to apply; default value.

=\*ALL

An attempt is made to output the message texts in the language defined for the task. A search is carried out first in the task-specific message files and then (if the result is negative) in the system message files. If the message text does not exist in the requested language, it is output in the standard language.

=\*TASK

Messages from the task-specific message files are only output if the message text exists in the language defined for the task. Otherwise, the message text in the standard language is output from the system message files.

=\*STD

Messages are output as defined in the JOIN entry or, if no definition is found there, as defined at system generation time.

### TASK-LANGUAGE

Serves to define a language to be used for message output. This definition applies to the current task run only.

=\*UNCHANGED

The language defined for the task run is not changed; default value.

=\*STD

The language specified in the JOIN entry or at system generation time is to be used.

=language

1 letter, where D = German, E = English. Symbols for other languages may be obtained from the "System Messages" manual [24] or the system administrator.

=\*NO

The language defined at system generation time as the language to be used for message output is used.



## MRSSTA Display multiprocessor system status

Application group: Multiprocessor systems (page 37)

### Command description

The MRSSTA command is only available with the software product MSCF (see the "MSCF Multiprocessor System" manual [15]).

The MRSSTA command displays on SYSOUT active and potential connections between the user's own (local) computer and other computers within the multiprocessor system network.

### Format and operand description

Operation	Operands
MRSSTA	[bcamname]

**bcamname** Specifies the name of the computer whose connection to the local computer is to be displayed.  
If this operand is omitted, information on all active and intended (potential) connections is output.

The display has the following format:

PROCESSOR	STATUS
bcamname	status
bcamname	BCAM name of the processor, as defined at generation of the communication system.
status	Current status of the processor "bcamname"
	MC-LOCAL "bcamname" is the local computer.
	MC-UNKNOWN "bcamname" is not the user's own computer and cannot be found in the table.
	MC-CONNECTED A connection to the computer "bcname" exists.
	NOT MC-CONNECTED There is no connection to the computer "bcname", but "bcname" is known to the local computer, i.e. a connection request from it would be accepted.

For **examples**, see the "MSCF Multiprocessor System" manual [15]).

**MSGCONTROL      Add or delete (task-specific) message files**

Application group: Job control (page 22 ff.)

**Command description**

The MSGCONTROL command enables you to add your own message files to the message system. These message files can be used for the current task only. They are accessed before the system message files during message searching. Up to 8 message files can be added and/or deleted with one MSGCONTROL command.

A message file comprises the message output and the corresponding HELP output (reduced message source file). The MSGEDIT utility routine must be used to create a message file and the MSGLIB routine must be used to divide it into the message work file and the HELP file. The names of the message work files are specified in the MSGCONTROL command (See "Utility Routines" manual [16]).

The SHOW-MSG-DEFAULT returns information on the names of the (system-wide) and (task-specific) message files.

## Format and operand description

Operation	Operands
$\left\{ \begin{array}{l} \text{MSGCONTROL} \\ \text{MC} \end{array} \right\}$	$\text{FILE} = ( [\text{ADD} = \left\{ \begin{array}{l} \text{file} \\ (\text{file}, \text{file}, \dots) \end{array} \right\} ]$ $[\text{, DEL} = \left\{ \begin{array}{l} \text{file} \\ (\text{file}, \text{file}, \dots) \end{array} \right\} ] )$

**FILE** Specifies the files to be added to or deleted from the message system on a task-specific basis.

=(ADD=file)

"file" = name of the message file to be added to the message system.

=(ADD=(file,file,...))

A list of up to 8 message files may be specified.

=(DEL=file)

"file" = name of the message file to be deleted from the message system.

=(DEL=(file,file,...))

A list of up to 8 message files may be specified.

## ON Initiate conditional execution of a command sequence

Application group: Job variable functions (page 36)

The ON command is only available with the software product JV (see the "Job Variables" manual [11]).

### Command description

The ON command allows you to make execution of a command sequence dependent on a particular condition.

An ON command sequence begins with ON and ends with ENDON. A condition may be specified in the ON command. Each time "condition fulfilled" occurs, a sequence of ENTER and/or SETJV commands is run (ON command sequence), if specified in the ON command sequence.

This applies until expiration of a timeout period, which must be additionally specified. It is also possible to specify in advance a maximum number of runs for the command sequence. In the case of "timeout", a similar sequence, also consisting only of ENTER and SETJV commands, is run, provided the maximum number of runs of the ON command sequence has not already been reached.

The conditions and command sequences are first stored and processing continues with the next command after ENDON. If the condition specified in the ON command is fulfilled, or if the timeout period has expired, the job is interrupted and the appropriate command sequence performed, unless the user is currently in the BREAK/ESCAPE mode. In the latter case, execution of the ON command sequence is put back to the RESUME or RTI command. In program mode (EXEC), the program is interrupted on completion of the command currently being executed. If the job is in the WAIT state, there is an immediate interrupt. In all other cases, the job is interrupted after completion of the current command. After the ON/TIMEOUT command sequence has been performed, execution is resumed at the point of interruption.

Any number of ENTER and/or SETJV commands may be specified in the ON/TIMEOUT command sequence. If an invalid command is used, an appropriate error message is output, and you can continue with the interactive entry of a valid command.

The ON command may be used in interactive and batch mode.

The ON statement sequence ends with the first command of the TIMEOUT statement sequence (marked with .name), or with the ENDON command if no timeout statement sequence was specified. If there is a TIMEOUT statement sequence, it will be terminated with the ENDON command.

If the ENDON command has been marked with the TIMELAB operand, in the event of a timeout an appropriate message is simply output via SYSOUT.

Upon completion of a procedure, any ON constructs which have been set up will remain effective instead of being erased.

The three possible combinations shown below will produce a warning or error message (with a branch to the next STEP command):

- ON statement sequence specified, cond.exp not specified:  
Warning: "ANGEGEBENE <ON-ANWEISUNGSFOLGE> WIRD ÜBERGANGEN".  
("specified ON statement sequence skipped")
- COUNT operand specified, cond.exp not specified:  
Warning: "ANGEGEBENER ZAEHLER IST NICHT WIRKSAM".  
("specified counter is not operative")
- TIMELAB operand specified, TIMEOUT statement sequence not specified:
  - in the batch and procedure modes:  
error message: "<TIMEOUT-ANWEISUNGSFOLGE> NICHT VORHANDEN"  
("TIMEOUT statement sequence not present")  
and error handling (branch to next STEP command).
  - in interactive mode:  
warning: "TIMEOUT-ANWEISUNGSFOLGE" IST EINZUGEBEN!!"  
("enter TIMEOUT statement sequence")

An ON command is deleted in the following circumstances:

- The ON command sequence has been performed the number of times specified in the COUNT operand.
- The period specified in the TIME operand has expired, and the TIMEOUT statement sequence, if one exists, has been executed.
- A DELON command has been issued for this ON command.
- The LOGOFF command has been issued (i.e. an ON command within a procedure may still be in the system after completion of the procedure).
- A job variable contained within "cond.exp" has been deleted.
- The catalog containing a job variable concerned is subject to long-term export.

## Conditional expressions

Conditional expressions forming components of the conditional statements described later must be composed according to the following conventions:

<cond.exp>	::= <cond.exp><logic.op><cond.exp>   <relation.exp>   (<cond.exp>)   NOT <cond.exp>
<logic.op>	::= AND   OR   XOR   NOT
<relation.exp>	::= <term> <comp.op> <term>
<term>	::= <jv-identifier>   <jv-substring-def>   <const>
<comp.op>	::= < >   =   <=   >=   <>   LT   GT   EQ   LE   GE   NE
<jv-identifier>	::= jvname   jv-link-name   special-jv-name
<jv-substring-def>	::= (<jv-identifier><jv-param-list>)
<jv-param-list>	::= <start-pos><length>   -
<start-pos>	::= ,positive integer   , Default value = 1;
<length>	::= ,positive integer   ,   - Default value = 64 or less if job variable value shorter. Maximum value = 256.
<const>	::= Any character constant or hexadecimal constant with a length between 1 and 64, for example: C'HALLO' or 'HALLO' or X'00FF' or X'0FF'

The syntax described above is supplemented by the customary rules of precedence governing logical operators. In other words, NOT is the most binding, followed by AND, then OR and finally XOR, which is the least binding. For instance, the following expression:

```
NOT JV1=C'ABC' OR JV2=C'Z' AND JV3<>JV4
```

is interpreted as

```
(NOT JV1=C'ABC') OR (JV2=C'Z' AND JV3<>JV4)
```

It is recommended that <relation.exp> always be enclosed in parentheses, e.g.:

```
(NOT JVA<>'RUN'), instead of NOT JVA<>'RUN'.
```

As regards the character-string nature of job variable values and the way they are handled in the system modules for the evaluation of conditions, certain restrictions should be noted:

- The meaningful length of job variable values within conditional expressions is limited to a maximum of 256 bytes.
- The evaluation of relational expressions is based on the character type of JV values, and corresponds to alphabetical order (EBCDIC: numerics are "greater" than alphabets).

*Example:*

The following expressions are "true":

C' ' < C'A'	
C'A' < C'B'	X'0123' < X'0124'
C'B' < C'BB'	X'00' < X'000'
C'GUTEN ABEND' < C'GUTEN MORGEN'	
C'ZZZZZZZZZ' < C'0' (!)	X'F0F0F0F0' < X'F1'
C'8' < C'9'	
C'899999999' < C'9' (!)	X'3FFF' < C' '

- A conditional expression is rejected in the following cases:
- A JV which has been referenced cannot be accessed.
- A JV which has been referenced cannot be found in the specified catalog.
- You are not authorized to access the JV, which is protected by a password or is not shareable.

## Execution logic

The execution logic of an ON command may be illustrated as follows:

```
DO WHILE NOT TIMEOUT
    IF <cond.exp> = 'TRUE'
        THEN
            - COUNT:=COUNT - 1
            - message on SYSOUT: '/ON: CONDITION=TRUE,COUNT=<counter>'
            - execution of the ON statement sequence, if present

        IF COUNT > 0
            THEN
                - no action
            ELSE
                - delete ON command
                - message on SYSOUT:

                'ON---<onid>-'
                '/ON terminated'

        ELSE
            - no action

    - continuation of processing
END;

TIMEOUT:
    - delete ON command
    - message on SYSOUT: '/ON':TIMEOUT,COUNT=<counter>'
    - execution of TIMEOUT statement sequence, if present
    - continuation of processing
```



## Error exit

If the command cannot be executed, (error detected during syntax check or upon initial evaluation of conditional expression), a branch is carried out to the next STEP command.

If the command sequence contains a branch back (SKIP/SKIPJV), the same ON may be set up a number of times. If you wish to prevent this, you must skip ON the next and subsequent times you encounter it. This can be achieved by means of a conditional branch, the condition for which is always satisfied after the initial setting up of ON, regardless of any flag. If an ON command sequence bears a flag specified for another ON which is still effective, then a warning message is output via SYSOUT and the ON command sequence is set up with this flag.

The ON command sequence is to be regarded as a contiguous unit, i.e.:

- It is not possible to branch into an ON command sequence from outside.
- Whenever a branch is to be performed (i.e. following a SKIP, WAIT, WHEN or, in the case of an incorrectly executed command, following a branch to LOGOFF, ENDP or the next STEP), a search is first performed for the accompanying ENDON for each ON encountered; only after this is the normal search for a specified flag or for STEP initiated.

## Format and operand description

Name	Operation	Operands
[.marker]	ON	[cond. exp.] [ , TIME=time] [ , TIMELAB= .name] [ , COUNT=number]

Immediately after issuing the ON command, specify:

[ON-statement sequence]  
[timeout-statement sequence]

.....  
/ENDON

.flag	Serves to identify the ON command. "flag" consists of up to 8 alphanumeric characters, the first of which must be alphabetic. Even if the user does not specify "flag", an ON identifier "onid" will be generated internally, provided that the ON construct is syntactically correct; "onid" is output via SYSOUT. This has the particular result that an interactive user can erase an ON statement, even if he has not specified a flag.
cond.exp.	see above, "Conditional expressions". "cond.exp" must not contain any special job variables. If "cond.exp" is not present, only the timeout specifications are effective.
COUNT=number	Specifies the maximum number of times that the ON statement sequence is to be executed: $1 \leq \text{number} \leq 32767$ . Default value = 1 The COUNT value is decremented by 1 each time "condition fulfilled" occurs.
TIME=time	Specifies time period in seconds:  Default value: 600 secs Maximum: 32767 secs Minimum: 1 sec Accuracy: 200 msec  If the period specified in the TIME operand has expired (measured from the initiation of the ON statement), the ON statement sequence itself will no longer be executed if "condition fulfilled" once more applies and the COUNT operand still has a value greater than 0.
TIMELAB=.name	Label of the first command in the TIMEOUT statement sequence. "name" consists of up to 8 characters. The first character must be a letter.

Refer to the "Job Variables" manual [11] for examples.

## OPTION Specify logging during job execution

Application group: Job control (page 22 ff.)

### Command description

The OPTION command enables you to define the type of logging and to output memory dumps while a job is executing.

By specifying the operand MSG, you can:

- choose between the standard and abbreviated forms of system messages on system file SYSOUT,
- log console messages on SYSOUT,
- have an additional log output to system file SYSLST.

The OPTION command controls the logging mode as specified in the LOGON or ENTER command by means of the MSG operand.

By specifying the DUMP operand, you can:

- suppress all memory dumps,
- output all memory dumps,
- suppress or output memory dumps optionally as they arise (in interactive mode only).

The dump is a user dump created by the CDUMP macro (see "System Exits" manual [10] or "Executive Macros" manual [5]).

The compact dump is output to disk as an unedited PAM file. It includes the task's entire class 5 and class 6 memory allocation as well as its system tables from class 3 and class 4 memory.

The file containing the dump is created under the ID of the user requesting the dump. As soon as the dump has been created, the system displays the message "DUMP WRITTEN,FILENAME=\$userid.DUMP.tsn.i" and the dump's title line.

"i" represents the dump sequence number if more than one dump is requested for each "TSN".

The file can be analyzed with the aid of an editing program (see "System Exits" manual [10]).

At least 1 command operand must be specified.

Logical DMS is used for compact dump output. Any error encountered in DMS will prevent dump output.

If an error occurs during user dump output, the dump is aborted. The error code is shown with the following message:

DMS xxxx ERROR CAUSED TERMINATION OF DUMP PROCESSING



Remove files to external storage or delete them if they are no longer needed. Dump files take up a great deal of space.

If there is not enough disk space available (less than 3 PAM blocks), the dump output is aborted and the message

INSUFFICIENT DISKSPACE:NO USERDUMP OUTPUT is displayed.

**Format and operand description**

Operation	Operands
$\left. \begin{matrix} \{ \text{OPTION} \} \\ \{ \text{OPT} \} \end{matrix} \right\}$	$[\text{MSG} = \left\{ \begin{matrix} \text{F} \\ \text{C} \end{matrix} \right\}] [\text{L}] [\text{H}] [\text{T}] [ , \text{DUMP} = \left\{ \begin{matrix} \text{STD} \\ \text{YES} \\ \text{NO} \end{matrix} \right\}] [ , \text{MAXLST} = \text{number}] [ , \text{MAXOPT} = \text{number}] [ , \text{TESTPRIV} = (m, n) ]$

**DUMP** Specifies whether the system, following the message "PROCESSING INTERRUPTED AT...", is to suppress, print or offer the user memory dumps if a cause for DUMP is encountered during the job run. At system generation time, an initial value for DUMP is specified (usually DUMP=STD). The initial value applies until otherwise specified by you.

**=STD** In interactive mode, the system asks:

DUMP DESIRED? REPLY (Y=YES, N=NO)




If you respond with "N", the memory dump is suppressed. If "Y", the system outputs the dump and issues the message:

DUMP IN PROGRESS, PLEASE WAIT

In batch mode and in procedures, DUMP=STD causes the dump to be suppressed and the following message to be issued: SYSTEM REGULATIONS PROHIBIT DUMP

**=YES** Every user dump is output and the following message issued on SYSOUT: DUMP IN PROGRESS, PLEASE WAIT

**=NO** Every user dump is suppressed and the following message issued on SYSOUT: DUMP PROHIBITED BY OPTION COMMAND


- MAXLST** Indicates the maximum number of records which the job can output (in summary form) to the system files SYSLST, SYSLST01, SYSLST02 through SYSLST99. SYSOUT records which are written simultaneously to SYSLST (MSG=FH option) are not counted.
- =number Number of records, where  $0 \leq \text{number} \leq 999999$ .
-  If the specified number is exceeded:
- In batch mode, the job is terminated abnormally.
  - In interactive mode, you can specify whether the job is to continue after "number" has been reached. If so, output again continues until "number" is reached.
- MAXOPT** Indicates the maximum number of records to be output for the job to the system file SYSOPT.
- =number Number of records, where  $0 \leq \text{number} \leq 999999$ .
-  If the specified number is exceeded:
- In batch mode, the job is terminated abnormally.
  - In interactive mode, you can specify whether the job is to continue after "number" has been reached. If so, output again continues until "number" is reached.
- MSG** Defines the type of logging for further job execution. At system generation time, a default value is specified for MSG (F or C). However, this default value is only effective if additional operand values are specified in the OPTION command for the MSG operand.
-  In some cases, owing to the internal execution sequence, the order in which messages are output does not correspond to the order of their causes.
- =F System messages are output in full-length form to the system file SYSOUT (F for "full message").
- =C System messages are output to SYSOUT in coded abbreviated form (C for "code").

=L Console messages and operator responses for this job are logged to SYSOUT. Operator commands controlling job execution (e.g. priority changes) as well as general (system-oriented) warnings and error messages for the operator are not output here (L for "log").


If MSG=LH is entered, messages logged to SYSLST will also be given the time of day at which they were issued.

=H All messages to SYSOUT are also logged to SYSLST (H for "Hold Message").

Exception: system messages requiring a response from you as user, and the message "ABNORMAL PROGRAM TERMINATION"


 In line mode (MODE=LINE), logging is performed line-by-line, i.e. NL control characters are evaluated. In format mode (MODE=FORM), logging is performed continuously, i.e. representation is not true to format; NL control characters are not analyzed.

=T In interactive mode, messages to SYSOUT are output to a hardcopy printer. Formatted messages and your entries are not printed.

 For output in line mode and with MSG=T, the whole message is regarded as one output unit (same output as for OHOM=YES in the WROUT and WRTRD macros).

TESTPRIV Defines a test privilege value (for debugging with AID).

=(m,n) m = read privilege, n = write privilege. "m" and "n" must not exceed the maximum values set in the JOIN entry (see SHOW-USER-ATTRIBUTES command, AIDR and AIDW fields. Initial values: m=1, n=1. When debugging a program, start with the initial values, which you can then increase up to the maximum values).

 The system administrator can use the JOIN entry to specify whether the system administrator's or operator's consent will be required for an increase in test privilege. If this increase is rejected, you receive an error message.

**Examples:***Example 1: Interactive mode*

```
/LOGON ... ,MSG=CHL _____ (01)
.....
```

```
/OPTION MSG=L _____ (02)
```

(01) The following set of conventions applies from the LOGON command until the first OPTION command:

- C Coded, abbreviated form for messages to SYSOUT;
- H Logging to SYSLST;
- L Console messages and operator responses logged to SYSOUT.

(02) The OPTION command modifies these conventions as follows:

- F Full-length form for messages to SYSOUT (default value);
- L Console messages and operator responses logged to SYSOUT.

*Example 2: Batch mode*

The command OPTION MSG=FH is issued in an ENTER file.  
SYSOUT log of the ENTER job:

```
/LOGON
/OPTION MSG=FH
/FSTAT X. ENTER.1
%0000003 :V:$PA123456.X. ENTER.1
%:V: PUBLIC:      1 FILE RES=          3, FREE=          2, REL=          0 PAGES
/OPTION MSG=F
/REMARK ENDE
/LOGOFF
% EXC0419 LOGOFF AT 0830 ON 90-08-29, FOR TSN XY95
% EXC0421 USED CPU TIME: 0.1770, SERVICE UNITS: 00000002156
```

From the LOGON command to the first OPTION command, the job run is only logged to SYSOUT. Then, the log is additionally written to system file SYSLST until the next OPTION command.

**SYSLST log**

```
(IN)  FSTAT X. ENTER.1
(OUT) 0000003 :V:$PA123456.X. ENTER.1
      :V: PUBLIC:      1 FILE. RES=          3, FREE=          2, REL=          0 PAGES
(IN)  OPTION MSG=F
```

Both system files are output to the printer after the ENTER job has terminated.

## PARAMETER      Control compilation/translation

Application group: Program control (page 30)

### Command description

The PARAMETER command and its operands can be used to control important compiler functions for the programming languages Assembler, COBOL, FORTRAN, ALGOL, PL1, RPG, etc.

Entries in the PARAMETER command allow you, for example , to

- assign macro libraries,
- create listings (e.g. compilation, error, diagnostic or cross reference listings)
- create the internal symbol dictionary (use of debugging aids).

The PARAMETER command is only supported for reasons of compatibility. The user is advised to use, instead of the PARAMETER command, the compilerspecific options in the COMOPT statements for the particular language processor involved.

The PARAMETER command must be entered before the language processors are called, i.e. prior to the EXECUTE or RESUME command.

If no PARAMETER command has been entered for a job, the default values are assumed for all operands.

The desired operands can either be specified in the same PARAMETER command or divided up over a number of separate commands. A second PARAMETER command will only modify those operands which are specified, and will not cause the other operands to revert to default values.

### Format

Operation	Operands
<pre>{ PARAMETER } { PARAM }</pre>	<p>The various operands in this command are described in detail in the reference manuals for the corresponding language processors and in the BS2000 "System Reference Guide".</p>



## PASSWORD      Specify password

Application group: File processing (page 26 ff.)

### Command description

The PASSWORD command is used to provide a job with passwords for opening files or for access to the catalog entry. For this purpose, a table of passwords is created which can be extended or reduced in stages, or also deleted, via subsequent PASSWORD commands. This password table is searched during processing of the CATALOG, FILE and ERASE commands, and also upon file opening, if file access requires a password not contained in the program's FCB (file control block). Access is allowed once the required password is found in the table. The password is replaced by the letter "P" when listed in a log. At the end of the job, the password table is deleted.

### Specify passwords

If both a write and a read password have been specified for a file, the write password assumes the function of both passwords, i.e. entering the write password allows both reading and writing.

If only a read password has been specified for a file, this password must be entered both for reading and writing. Any existing read password must be specified if the write password is modified.

If both a read and an execute password have been specified for a file, the read password assumes the function of the execute password, i.e. entering the read password allows both reading and executing.

**Remote File Access** (see also "RFA" manual [12]).

The PASSWORD command is automatically forwarded by the requesting job to all RFA partner jobs.

**Format and operand description**

Operation	Operands
PASSWORD	$\left[ \left\{ \begin{array}{l} \text{password} \\ (\text{password}, \dots) \end{array} \right\} \right] [ , \text{REL}=\text{YES}]$

**password** (file) password. Specified as a character ("C") string, hexadecimal ("X") string or decimal number.

C string: C'character-constant'; up to 4 characters.  
X string: X'hexadecimal-constant'; up to 8 characters.  
Decimal number: decimal-constant; valid range: -2147483648 through 2147483647.

The password with the value X'00000000' is ignored. Up to 63 passwords may be entered in one PASSWORD command.

**REL=Y[ES]** Specifies that the passwords defined in the PASSWORD command are to be deleted from the job's password table. If no password is specified in conjunction with the REL operand, the task's entire password table is deleted.

**Example:**

During an interactive job, input includes the following commands:

```

/LOGON ...

(IN)    PASSWORD X'51EF' _____ (01)
(IN)    CAT S.SAL.FILE1,STATE=U,RDPASS=C'OR' _____ (02)
(IN)    PRINT S.SAL.FILE1
(OUT)   % SCP0860 FILE PROTECTED BY A READ PASSWORD.
(OUT)   PRINT REQUEST REJECTED FOR :V:$PA123456.S.SAL.FILE1

(IN)    COPY S.SAL.FILE1,S.SAL.FILE1.KOPIE
(OUT)   % DMS05F3 REQUIRED PASSWORD IS NOT IN PASSWORD TABLE.
(OUT)   ENTER PASSWORD AND RETRY CMD LATER

(IN)    PASSWORD C'OR' _____ (03)
(IN)    PRINT S.SAL.FILE1
(OUT)   % SCP0810 PRINT :V:$PA123456.S.SAL.FILE1 ACCEPTED: TSN: 7710, PNAME: SALEM

(IN)    COPY S.SAL.FILE1,S.SAL.FILE1.COPY

(IN)    PASSWORD REL=Y _____ (04)
(IN)    ER S.SAL.FILE1
(OUT)   % DMS0801 ERROR WHEN DELETING FILE :V:$PA123456.S.SAL.FILE1
(OUT)   % DMS05BF FILE PASSWORD-PROTECTED. FIRST ENTER CORRECT PASSWORD
        VIA APPROPRIATE COMMAND, THEN REENTER /DELETE-FILE OR /ERASE.

(IN)    ER S.SAL.FILE1.COPY _____ (05)

(IN)    FSTAT S.
(OUT)   0000003 :V:$PA123456.S.SAL.FILE1
        :V: PUBLIC:      1 FILE. RES=      3, FREE=      2, REL=      0 PAGES

/LOGOFF

```

- (01) The first PASSWORD command in this job serves to set up its password table and enter the password X'51EF' in it. Subsequently, access is permitted to all files protected by this password.  
In the log, the password is overwritten by the character string "SS...S".
- (02) A read password is defined for the file S.SAL.FILE1 with the aid of the CATALOG command. The password is hereby entered in the catalog, but not recorded in the job's password table. The subsequent PRINT or COPY command leads to a corresponding error message.
- (03) The PASSWORD command enters the password C'OR' in the job's password table. The subsequent PRINT and COPY commands are therefore executed as requested.
- (04) This PASSWORD command causes the password table to be deleted; the following ERASE command is rejected.
- (05) The file S.SAL.FILE1.COPY can be deleted because it is not password-protected.

## PAUSE Inform operator and wait

Application group: Job control (page 22 ff.)

### Command description

The PAUSE command causes a message to be output on a main console (see the TYPE command), and the job to be halted until the operator has responded.

The PAUSE command can be entered in both batch and interactive mode.

The message is normally sent to the main console. If it starts with "<" (less than), the next character is interpreted as a routing code (see the "System Operator's Guide" [2]) and the message is sent to the appropriate destination.

Any operator response is not normally sent to SYSOUT. However, the user can have all console messages and operator responses concerning the job logged to SYSOUT by specifying MSG=L in the LOGON or OPTION command.



Comments are not permitted in the PAUSE command.

### Format and operand description

Operation	Operands
PAUSE	message

message            Message to be output to the console. It must not exceed 72 characters. All characters on the keyboard are permissible.

## PRINT Output (print) file

Application groups:      Job control (page 22 ff.)  
                              File processing (page 26)  
                              Device control (page 33)  
                              SPOOL jobs (page 35)

### Command description

The PRINT command allows you to output files to printer, batch terminal or magnetic tape. Unless otherwise specified, output is directed to a local printer (line or laser printer). The output medium is determined by The FORM/LOOP/CHARS/CHARS-POOL/FOB/ROTATION operands specified in the PRINT command or the default values for these operands.

If output is to a laser printer, you may choose your own character sets and also change the character set at any desired point within the text. If output is to a HP laser printer, page rotation and text overlays can be specified if the appropriate hardware is provided - page rotation module or graphics memory.

The description is valid for SPOOL V2.5B / RSO V2.1B.

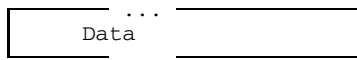
The specifications from the PRINT command, together with the definitions (default values) on device selection, are entered in a SPOOL control block (SCB). The SCB is added to one of the spoolout queues (local spoolout, RSO, RBP) and managed as a separate job with its own TSN. You can follow the progress of your spoolout job by using the STATUS command, change the run priority with the PRIORITY command, and cancel the job with the CANCEL command.

### Processing a spoolout job:

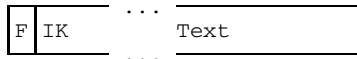
- A record is read from the file to be printed.
- One or more print lines are generated from the record (depending on the values specified for the operands FROM, TO, BINARY (STARTNO, ENDNO), SPACE).
- The requisite channel command words are created for each print line.
- An I/O operation is started for the print line (or, in the case of printers which have loadable buffers, for the print page).

In the case of output to a printer, line or page feeds can be controlled by the user.

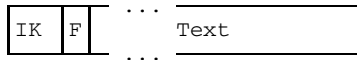
**Examples for the format of records in a file to be printed:**



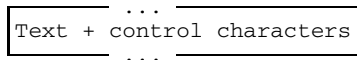
The record consists of printable characters only (without feed control characters).



The record begins with a feed control character (F), followed by text (including the ISAM key (IK) in the case of ISAM files).



The record begins with the ISAM key (KEYPOS=5), followed by the feed control character (F) and the text.



The record contains a combination of text and control characters (e.g. to change the character set).

**Use of character sets**

All character sets which you have selected (either explicitly or by specifying a pool name) are loaded on the desired output device at scheduling time.

The file format for PRINT file,....,CONTROL=PHYS is structured differently for HP and ND laser printers.

If CONTROL=NO is specified, only the first selected character set may be used. The number of character sets (selected explicitly with PRINT or in the specified pool) is compared with the upper limit defined as a global parameter at installation time.

The scheduler checks as to whether the specified device has sufficient fonts to enable it to load all the character sets selected (either explicitly or implicitly).

Wherever possible, use the pools created by the system administrator

- either by specifying the pool name only (CONTROL=PHYSICAL)
- or by specifying a pool name and an index (CONTROL=NO).

The scheduler groups jobs that require the same character set or the same pool, thus preventing unnecessary reloading of character sets.

If, in PRINT, a particular character set is specified that is not yet loaded in one of the device fonts, and the character buffer is also full, the controller searches the loaded character sets for the one most seldom used and replaces it with the newly-requested character set.

If more than four character sets are addressed in the PRINT command, there may be no accessible HP laser printer capable of loading the number of character sets requested. In such a case, the task in question cannot be processed.

For this reason, all tasks addressing more than four character sets (or FOBs, or the page rotation module) are marked with "\*" in the STATUS LIST in the OPT field. The number of character sets requested is output with STATUS TSN.

## Printer selection

If the character records and LOOP records specified in the PRINT command are contained in the \$TSOS.SPOOLFILE file, the printer selection depends on the printers available and the operator's specification.

If the COPIES=number2, CONTROL=PHYS, IMAGE or DIA operands are specified, output is to an ND/HP laser printer. If LOOP=(...), CHARS-POOL, ROTATION, FOB are selected, output is to an HP printer. The operator can also specify that laser printer output requested with CONTROL=NO may be output on mechanical printers. The LOOP, COPIES (format 2), IMAGE and DIA operands are ignored if a printout intended for a laser printer is output on a mechanical printer.

The printer used depends on the operands specified and is selected on a hierarchical basis as follows:

Class	Operand	Printer selection (minimum requirement)
A	DEVICE=REMOTE T= USER= DEFER=YES	Remote batch terminal
	DEVICE=device	RSO printer
B	IMAGE= CHARS= (format 2) COPIES= (format 2) DIA= CONTROL=PHYS	Laser printer, if no class A operand specified
C	LOOP=	Mechanical printer or laser printer if no class A or class B operand was specified

If no operands are specified from class A, B or C, the minimum requirement is any printer capable of printing 132 (or 136) characters per line.

**LOOP/IM/BI/FORM records for 3337, 3338, 3339, 3348 and 3365 Printer.**

The above printers are controlled by means of information stored in hardware buffers in the \$TSOS.SPOOLFILE file, which contains the following types of record:

- LOOP records for loading the vertical format buffer (VFB) or the format control buffer (FCB) for paper feed.
- IM or BI records for loading the image memory. The type chain character assignment is mapped to these records.

The printer buffers are loaded with default LOOP records from the \$TSOS.SPOOLFILE unless the FORM/LOOP operands are specified in the PRINT command. If the FORM=form operand (without LOOP) is used, the LOOP and IM (or BI) records are loaded into the buffer according to the specification for the FORM operand. The LOOP operand must be specified if the user requires a format control feature not declared in FORM. After the system administrator has entered an appropriate LOOP record in the \$TSOS.SPOOLFILE file, you can specify the name of this record at any time in the LOOP operand. If both the FORM and the LOOP operands are specified, the LOOP record specification contained in the LOOP command is used.

**Feed control characters**

SIEMENS feed control characters:

Feed control character	Effect
X'40' to X'4F'	Line feed before printing, line feed after printing
X'00' to X'0F'	Line feed after printing
X'C1' to X'CC'	Line feed before printing
X'81' to X'8C'	Line feed after printing

Channel 12 is reserved for SPOOL.



## Structure of SIEMENS and IBM printer control bytes:

Value	Bit position and significance (SIEMENS)							
	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
0	Line feed	After printing	Set to 0 by SPOOL	Set to 0 by SPOOL	Number of lines or channel number			
1	Line feed	Before printing						

Value	Bit position and significance (IBM)							
	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
0	Line feed	Channel number	Channel number	Number of lines	Number of lines	Always set to 0	After printing	
1	Form feed	Channel number	Channel number	Channel number	Channel number		Before printing	Always set to 1

ASA feed control characters and corresponding SIEMENS feed control characters:

ASA	SIEMENS	Effect
C'+'	X'00'	No line feed
C'-'	X'41'	2-line feed before printing, 1-line feed after printing
C'A'	X'CA'	Branch to channel 10 before printing
C'B'	X'CB'	Branch to channel 11 before printing
C'0'	X'40'	1-line feed before printing, 1-line feed after printing
C'1'	X'C1'	Branch to channel 1 before printing
.	.	.
.	.	.
C'9'	X'C9'	Branch to channel 9 before printing

Invalid control characters are corrected for laser printers.



Owing to differences in the printer hardware, ASA and Siemens feed control characters are not totally equivalent:

Siemens printers do not perform a line feed before printing, but always after printing, with the exception of X'00' (see above).

For more details, refer to the "SPOOL" manual [19].

### Outline description of the PRINT functions:

Operand	Function
pathname prefix[name] *	} Indicate files to be output.
eamnr *SYSLST *SYSLSTn *SYSOUT	
*L-E *LIBRARY- ELEMENT	} Indicate library members to be output.
*LIB	
START-SPOOL	Only for system files; specifies the time of printing.
ERASE	Erases printed file after printing (positional operand)
DESTROY	Overwrites the file with binary zeros (positional operand).
DELETE-FILE	Deletes file or overwrites it with binary zeros after printing.
FAMILY	Specifies whether a communal TSN should be assigned if a number of files are to be output.
LOCK	Edit-protects file to be output until end of output or end of job.
RETPD	Defines a retention period for the file.
PAGECC	Indicates whether the file to be printed contains page control characters.
PRIORITY	Declares a priority for the spoolout job.
PNAME	Declares a job name for the spoolout job.
TEXT	Allows specification of information for processing system exits.
CCPOS	Specifies the position of the control characters.

Operand	Function
CHKPT	Specifies whether the checkpoint is to be set on each logical page or on SECTION records.
COPIES	Determines how many additional copies of the file are to be printed; for laser printers also specifies how often each page printout is to be repeated successively.
DEVICE	Denotes the type of printer and spoolout job (e.g. local spoolout or RBP).
DUSER	Specifies the user ID under which the job is to be executed.
DACCOUNT	Specifies the account number for the user ID specified with DUSER.
DPASSWD	Specifies the password for the user ID specified with DUSER.
PRINTER	No longer used.
DEFER USER T	For output to batch terminal only: Determine recipients of the printout, and terminal ID.
FORM LOOP	Indicates print paper (type of form) and LOOP record.
SECTION	Restricts output to an area delimited by specifiable character strings.
SPACE	Determines the number of line feeds or the type of feed control characters.
HEADER	Declares a header line for every print page.
LINES	Determines the number of lines per print page.
FROM TO	Restricts output to a particular number of print pages or print lines.
STARTNO ENDNO	Restrict output to a specific subarea of the record to be printed.
SHIFT	Shifts the output text to a specific print column.
HDRNUM	Selects various computer-center-specific header pages.

Operand	Function
TRLNUM	Selects various computer-center-specific trailer pages.
BINARY	The data is also printed out in hexadecimal notation.
TAPE	Allows file to be output to tape.
DEVIN	Indicates that the file to be output is stored on tape.
VOLUME	Only for output to tape: Specifies the tape volume serial number.
CHARS	Designates character sets from a character set file (NDFILE or HPFILE; only for laser printers or RSO devices).
CHARS-POOL	Allows the use of a character set pool with up to 64 characters or of a specific character set from the specified character set pool for an HPFILE (only for HP laser printers).
CHARS-MODIFICATION	For RSO output; allows suppression of character set attributes except for CHARACTER-TYPE, LANGUAGE and NEAR-LETTER-QUALITY.
FOB	Allows print page overlay with text/graphics not contained in the file to be printed (only for HP laser printers with graphics memory).
RESOURCE	Specifies the resources to be used with the LP65 Printer.
ROTATION	Allows page rotation: the print page set up in the printer is rotated a certain number of degrees for printout onto paper (only for HP laser printers with page rotation module).
CONTROL	Determines the type of control character analysis (only for laser printers and RSO devices).
IMAGE	Denotes a user NDFILE, HPFILE, or RSOFILE. A user NDFILE for 3350/3352 laser printers may contain LOOP records and character set records. A user HPFILE for 3351/3353 laser printers may contain LOOP, character set, FOB and POOL records. A user RSOFILE for RSO printers may contain LOOP records and TRANSLATION-TABLES.
DIA	Determines which forms overlay negative is to be used (for laser printers only).

Operand	Function
DESTINATION	Determines an RSO printer or a pool of RSO or local printers to be used for output.
TRANSLATION-TABLE	Defines a code conversion table (and, where applicable, the user RSOFILE in which it is stored) to be activated for the duration of spoolout.
TRAY	Specifies the number of the tray from which the print paper is to be taken (for RSO devices only).
TRUNC	Indicates whether the file printed with "ERASE" specified is to be deleted even if an error occurs during printing.

## Format and operand descriptions

Operation	Operands
<pre>{PRINT} {PR }</pre>	<pre>{   pathname   (pathname,...)   prefix[name]   eamno   (eamno,...)   *   {     *SYSLST     *SYSLSTn     [*SYSLSTn,...]   } [, START-SPOOL={     [IMMEDIATE]     n     CLOSE     NO   } ]   *SYSOUT [, START-SPOOL={     [IMMEDIATE]     CLOSE     NO   } ]    {     *L-E     *LIBRARY-ELEMENT     *LIB   } (LIBRARY=libname, ELEMENT=member name    ,TYPE=type [,VERSION={     *HIGH     version   } ]    [, CREATION-DATE={     LATEST     date     date(time1,time2)     RANGE(date1,date2)   } ]    [,RECORD-TYPE= (     FROM=number     TO=number   ) )</pre>

Operation	Operands
<pre>{PRINT} {PR (cont.)</pre>	<pre>       {ERASE       DESTROY       } [, {   DELETE-FILE={NO                ERASE                DESTROY}       }       {STD       YES       NO       } [, TRUNC={STD           IGNORE           KEEP       }       {device       pool       }       {RESOURCE=LP65 (PAGEDEF=number, DIAR={number       0       }, DIAV={number       0       },       INTRAY=number, OUTRAY=number, DUPLEX={STD       NO       YES       TUMBLE       })       }       [, CCPOS=number]       {STD       NO       }       [, PAGECC={       }       [, DUSER=duser]       [, DACCOUNT=daccount]       [, DPASSWD=dpasswd]</pre>



Operation	Operands
<pre>{PRINT} {PR (cont.)</pre>	<pre>[,SECTION=( [ID=string, POS=pos]               [,FIRST={number                         {string(, POS=pos, NUMBER=number)} } ]               [,LAST={number                        {string(, POS=pos, NUMBER=number)} } ] )  [,CHKPT={ONPAGES}         {SECTION} ]  [,IMAGE={name         {filename} ]</pre>
<pre>----- Output</pre>	<pre>[,LOCK={YES}         {NO} ]</pre>
<pre>to</pre>	<pre>[,RETPD=days]</pre>
<pre>printer,</pre>	<pre>[,PRIORITY=p]</pre>
<pre>batch</pre>	<pre>[,TEXT='text']</pre>
<pre>terminal</pre>	<pre>[,COPIES=number]</pre>
<pre>or</pre>	<pre>[,DEVICE={ {CENTRAL}             { *CENTRAL }             REMOTE } (DEVICE-TYPE={ *ALL                                      ND                                      HP } ) ] [,PRINTER=136]</pre>
<pre>tape</pre>	<pre>[,DEFER={NO} ] [, {USER=userid2}              {YES} ] [, {T=statid} ]</pre>

Operation	Operands
$\left\{ \begin{array}{l} \text{PRINT} \\ \text{PR} \end{array} \right\}$ (cont.) <i>Output</i>	$[, \text{FORM} = \left\{ \begin{array}{l} \text{STD} \\ \text{form} \end{array} \right\}] [, \text{LOOP} = \text{loop1}]$
<i>to</i>	$[, \text{SPACE} = \left\{ \begin{array}{l} \frac{1}{2} \\ 2 \\ 3 \\ \text{E} \\ \text{A} \\ \text{I} \end{array} \right\}]$
<i>printer,</i>	$[, \text{HEADER} = \left\{ \begin{array}{l} \text{NO} \\ ([\text{T}] [ , \text{D}] [ , \text{P}]) \\ \text{S} \end{array} \right\}]$
<i>batch</i>	$[, \text{LINES} = \text{lines}]$
<i>terminal</i>	$[, \text{FROM} = \left\{ \begin{array}{l} \text{m} \\ -\text{m} \end{array} \right\}] [ \left\{ \begin{array}{l} \text{P} \\ \text{L} \end{array} \right\} ] [, \text{TO} = \text{n} [ \left\{ \begin{array}{l} \text{P} \\ \text{L} \end{array} \right\} ] ]$ $[, \text{STARTNO} = \text{byteno}] [, \text{ENDNO} = \text{byteno}]$
<i>or</i>	$[, \text{SHIFT} = \text{lines}]$
<i>tape</i>	$[, \text{HDRNUM} = \text{n1}]$ $[, \text{TRLNUM} = \text{n2}]$ $[, \text{BINARY} = \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}]$ $[, \text{TAPE} = \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \\ \text{TAPE-C1} \\ \text{T1600} \\ \text{T9P} \\ \text{T6250} \\ \text{T9G} \end{array} \right\}]$ $[, \text{DEVIN} = \text{TAPE}]$ $[, \text{VOLUME} = \left\{ \begin{array}{l} \text{SCRATCH} \\ \text{vsn} \\ (\text{vsn}, \dots, \text{vsn}) \end{array} \right\}]$ $[, \text{CONTROL} = \left\{ \begin{array}{l} \text{NO} \\ \text{PHYS} \end{array} \right\}]$

Operation	Operands
$\left\{ \begin{array}{l} \text{PRINT} \\ \text{PR} \end{array} \right\}$ (cont.)	$[, \text{LOOP} = \left\{ \begin{array}{l} \text{loop1} \\ (\text{loop1}, \text{loop2}) \\ (, \text{loop2}) \end{array} \right\}]$
<i>Output</i>	$[, \left\{ \begin{array}{l} \text{CHARS} = (\text{c1} [, \text{c2} [, \text{c3} [, \text{c4}]) \\ \text{CHARS-POOL} = (\text{pool} [, \text{number}]) \end{array} \right\}]$
<i>to</i>	$[, \text{FOB} = \text{fob}]$
<i>laser</i>	$[, \text{ROTATION} = \left. \begin{array}{l} \text{NO} \\ \text{YES} \\ 90 \\ 180 \\ 270 \\ (0, 180) \\ (90, 270) \\ (180, 0) \\ (270, 90) \end{array} \right\}]$
<i>printer</i>	$[, \text{IMAGE} = \text{image}]$ $[, \text{DIA} = \text{cc}]$ $[, \text{CONTROL} = \left\{ \begin{array}{l} \text{NO} \\ \text{PHYS} \end{array} \right\}]$ $[, \text{COPIES} = (\text{number1}, \text{number2})]$
-----	-----

Operation	Operands
<pre>{PRINT} {PR (cont.) Output to RSO devices</pre>	<pre>[,DEVICE=device]  [,CHARS=(c1[,c2]...[,c16])]  [,CHARS-MODIFICATION={YES NO}]  [,CONTROL={NO LOGICAL LINEMODE PHYSICAL TRANSPARENT}]  [,SPACE={1 2 3 NO E A I}]  [,TRANSLATION-TABLE=(name1[,name2])]  [,TRAY=number]</pre>

## Positional operands

pathname	stands for [ :catid: ] [ \$userid1. ]	$\left\{ \begin{array}{l} \text{file} \\ \text{group} \left\{ \begin{array}{l} (*\text{abs}) \\ (+\text{rel}) \\ (-\text{rel}) \end{array} \right\} \end{array} \right\}$
(pathname,...)	When more than one path name (up to 16) is specified, they must be separated by commas and enclosed in parentheses. The maximum number of characters which can appear in the parentheses is 256.	
catid	Catalog ID of the subset in which the file is stored. Default value: the catalog ID assigned to the user ID; (JOIN entry).	
userid1	User ID which is assigned to the file. Default value: the user ID from the LOGON command.	
group	Name of a file generation group from which a file (generation) is to be output. The name length must not exceed 34 characters and the name must not be partially qualified (see the "DMS" manuals [8,9]).	
abs	Absolute generation number of the file generation to be printed. "abs" is a member of the set (1, 2, ..., 9999).	
rel	Relative generation number of the file generation to be printed. "rel" is a member of the set (1, 2, ..., 99). "rel" is relative to a base value (see the BASE field in the FSTAT command output). The following applies: rel = "abs" - base value.	
file	Fully or partially qualified file name, the fully qualified name of a file generation, or the name of a temporary file. PAM files must not be specified in batch mode.  The PRINT command is rejected if the specified file <ul style="list-style-type: none"> <li>– is a newly cataloged file to which no data has yet been written;</li> <li>– has already been opened in output mode.</li> </ul> If the file is not cataloged under your ID, it must be shareable (see the CATALOG command).	

The following points must be borne in mind in conjunction with the output of cataloged files:

- A spoolout job will be created even if the file to be output has been reserved using a SECURE command. However, the reservation must have been lifted by the time the spoolout job is running, otherwise the job will not be executed.  
The file to be output will remain locked until the end of the session if the operand LOCK=YES has also been specified in the PRINT command and the job could not be executed due to this reservation.
- If the file to be output belongs to a file generation group ("group" specification in the path name), the ERASE operand is ignored.
- The EAM file \* and the system files SYSLST and SYSOUT may be combined as desired in a PRINT command, e.g. PRINT (\*,\*SYSLST,\*SYSOUT), but they must not be combined with a cataloged file or an EAM file number.
- The name of an output file generated with TAPE may be up to 54 characters in length. This must be taken into account when the input file name is assigned.
- In the case of foreign files on multifile tapes, only the first file is output.

prefix[name]

The specified temporary (user) file is output.

If only "prefix" is specified, then all the temporary (user) files are output.

"prefix" = the character which was specified at the time of system generation as the prefix to the names of temporary files.

"name" = any (file) name; length ≤ 30 characters.

The following points must be borne in mind when temporary files are output:

- A PRINT command for a temporary file is always carried out automatically with LOCK=YES,ERASE. This prevents a temporary file from being erased prematurely with LOGOFF, erasure not taking place until after print termination.
- The temporary file is also erased if the spoolout job is terminated abnormally (e.g. by the CANCEL command).
- For temporary files, see the manual "DMS Introductory Guide and Command Interface" [8].

eamno	Number of an EAM file, specified in the form of a decimal number.
(eamno,...)	If several numbers are specified, they must be separated by commas and enclosed in parentheses. See the "DMS Introductory Guide and Command Interface" manual [8] for more information.
*	<p>Specifies the EAM object module file of the current job.</p> <p>The following points must be borne in mind in conjunction with the output of EAM files:</p> <ul style="list-style-type: none"> <li>– EAM files (identified by file number or *) are deleted at the end of output. The files are also deleted if the spoolout job is terminated abnormally (e.g. by the CANCEL command). Up to 2268 EAM files may be specified in one PRINT command.</li> <li>– The EAM file * and the system files SYSLST and SYSOUT can be combined in a PRINT command in any way, e.g. PRINT (*,*SYSLST,*SYSOUT). However, they must not be combined with an EAM file number or a cataloged file.</li> </ul>
*SYSLST	Outputs system file SYSLST. If assignment is to a cataloged file, the latter's contents will be output.
*SYSLSTn (*SYSLSTn,...)	Refers to the system file SYSLSTn, where "n" is a (two-digit) number in the range (01,02,...,99). A maximum of 11 different SYSLST files can be specified in one list (enclosed in parentheses and separated by commas). All operands which have been specified are then valid for each of these system files.
*SYSOUT	<p>The system file SYSOUT will be output. If it is assigned to a cataloged file, the latter's contents will be output. This operand may only be specified in batch mode.</p> <p>The following points must be borne in mind in conjunction with the output of system files:</p> <ul style="list-style-type: none"> <li>– System files SYSLST and SYSOUT are deleted at the end of output. The files are also deleted if the spoolout job is terminated abnormally (e.g. by a CANCEL command).</li> <li>– The EAM file * and the system files SYSLST and SYSOUT can be specified in a PRINT command in any combination, e.g. PRINT (*,*SYSLST,*SYSOUT). However, they must not be combined with an EAM file number or a cataloged file.</li> <li>– The default values both for EAM files and for SYSLST and SYSOUT are SPACE=E and ENDNO=2048.</li> </ul>

- START-SPOOL** Only for system files; specifies the time of printing. All the operand values in a PRINT command which includes START-SPOOL=*n* or START-SPOOL=CLOSE are recorded for application to each of the system files addressed.  
This START-SPOOL specification can only be changed via the intermediate step START-SPOOL=NO. Otherwise the PRINT command will be rejected with an error message.  
Every print job is given a separate TSN.
- =IMMEDIATE** The print order is issued immediately; default value. IMMEDIATE cannot be specified for tape files or temporary files.
- =*n*** Printing is started after every "*n*" logical pages;  $n \geq \text{MIN}$ . The residual contents of the file concerned ( $\leq n$  pages) are printed after the system file has been closed (see the CLOSE operand).  
The minimum value MIN for "*n*" can be displayed using the SHOW-SPOOL-PARAMETERS command, the information being taken from the output field SPOOLOUT-OPTIONS:...ST-SP-LOW-VAL=.... "*n*" is not possible for tape files or temporary files.
- =CLOSE** The spoolout job for the system file is started as soon as the system file has been closed.  
A system file is closed:  
a) in the case of a primary allocation, with the LOGOFF command,  
b) if the allocation is to a cataloged file:
  - by entry of a new SYSFILE command (to alter the assignment) for the same system file
  - using the LOGOFF command (note: LOGOFF NOSPOOL suppresses the output)
  - when procedure step 0 is reached during procedure execution.
- =NO** Any previous specification of "*n*" or CLOSE is cancelled. If a value has been specified for the operand "*n*", then the printout job for the residual contents begins immediately after START-SPOOL=NO. Any further operands are rejected.



$$\left. \begin{array}{l} *L-E \\ *LIBRARY-ELEMENT \\ *LIB \end{array} \right\} (\dots)$$

A member from a PLAM library is to be printed

**LIBRARY=name** Name of the PLAM library.

**ELEMENT=name** Name of the PLAM library member to be printed. Wildcards are permitted in "name". Only one name may be specified.

**VERSION** Version of the member to be output.

**=\*HIGH** The highest existing version in alphabetical order is to be output; default value.

**=version** Version name of the member to be output. "version" may contain up to 24 alphanumeric characters; if wildcards are used, up to 40 alphanumeric characters.

**TYPE** Type of the library member to be output.

**=name** "name" may comprise a single letter or, if wildcards are used, a maximum of 12 letters.

**CREATION-DATE**

Creation date that serves as a criterion for selecting library members.

**=LATEST** Selects the most recently created library member for output.

**=date** Selects library members created on the specified calendar date for output.

**=date(time1,time2)**


Selects library members created on the specified date and within the specified time interval for output.

Default value for "time1": 00:00:00 hours.

Default value for "time2": 23:59:59 hours.

**=RANGE(date1,date2)**

Selects library members created within the given period (specified by calendar dates) for output.

- ERASE Erases the file as soon as output is terminated but only if you have write access to the file.  
Default value: the file is not erased after printout.  
If the file to be output belongs to a file generation group ("group" specification in the path name), the ERASE operand will be ignored.
- DESTROY Specifies that the catalog entry and the data in a file are to be overwritten with X'00...0' after printout. (This does not apply to EAM files and cataloged system files.)  
Default value: the file is not overwritten after printout.
-  ERASE and DESTROY are positional operands; command format must be observed!  
Neither ERASE nor DESTROY may be used in combination with the \*SYSLST, \*SYSLSTn or \*SYSOUT operands.  
If the file to be printed has the catalog entry ACCESS=READ, SPOOL automatically resets the ERASE or DESTROY operand.
- DELETE-FILE
- =NO The file is not deleted after printing (exception: EAM files and system files); default value.
  - =ERASE Same function as for ERASE
  - =DESTROY Same function as for DESTROY

## Keyword operands (in alphabetical order)

### BINARY

=NO

Operand for output to printer, batch terminal or tape. Outputs in character format only. Records extending beyond the print line are truncated; default value.

=YES

Outputs records in character format and in hexadecimal format.

The output format is determined by the output destination:

#### Local printer

The format of the output depends on the line length defined for the forms used. The beginning of each output line contains an 8-byte prefix, followed by data with the following length:

line length (according to form definition) minus 8 bytes.

Each output line is printed first as per character set and then repeated in hexadecimal form.

#### *Format of the output line*

Column	Contents
1-4	Number of the first byte in the record to be output in this line.
5-8	Blanks.
From 9 onward	Characters of the record as per character set, the individual characters being separated by a blank. The next line repeats the characters in hexadecimal format.

#### Batch terminal

Each input record is distributed in units of 32 bytes over several output lines.

**Format of the output line**

Column	Contents
1-4	Number of the record. This number appears only in the first output line of each input record.
5-14	Blanks.
15-18	Number of the first byte output, relative to the beginning (=0) of the record.
19-22	Blanks.
23-86	32 characters (hexadecimal) of the record.
87-92	Blanks.
93-124	Characters represented as per character set. Non-printable characters are replaced by blanks.

CCPOS=ccpos

Specifies the position within the data records of the feed control characters interpreted with SPACE=E//A for printout. The record length field is not taken into account for records of variable length.

CHARS

**1.) Output to laser printer;**

CHARS designates one or more character sets to be used for printing the file.

=(c1,c2,...)

The specifications "c1, c2, c3, c4" each give the name of a character set. The character sets must be in the file \$TSOS.NDFILE or \$TSOS.HPFILE, or in an NDFILE/HPFILE created by the user (IMAGE operand). (The name of the first character set and the number of specified character sets are displayed in the output for the command STATUS tsn.)

Use of more than one character set requires the specification CONTROL=PHYS. When CONTROL=NO is specified, only the first character set specified is used to print the (complete) file.

The default value for CHARS specifying the form being used can be interrogated using the SHOW-SPOOL-FORMS command; the information is contained in output field C-S (CHARACTER-SET).



The header line for each page (HEADER operand) is printed with the first character set specified.

In the case of HP laser printers, the system administrator can specify whether the header page is printed using the standard character set or the one specified under CHARS or CHARS-POOL. The setting can be interrogated using SHOW-SPOOL-PARAMETERS; this information is contained in the output fields HEADER-PAGE and CHARACTER-SET.

**2.) Output to RSO printers:** CHARS specifies up to 16 character sets that are to be used to print the file. If the CHARS operand is omitted, the character set assigned to the specified form is used.

=(...)

List of the character set names.

If the CONTROL=LOGICAL operand is omitted, the first character set specified is used for printing, otherwise the character set specified in the character set identifier (CSI) is used for printing for each line.

Use of the CHARS operand enables records to be printed using different character sets.

## CHARS-MODIFICATION

Operand for output to RSO printers; defines whether all or only selected character set attributes are to be taken into account for the print job. These attributes include, for example, font, near-letter quality printing (NLQ), color, etc. (see also the SHOW-SPOOL-CHARACTER-SETS command in the "SPOOL System" manual [19]). This operand has no effect for header and trailer pages.

=YES

All attributes of the character sets used for printing are taken into account for this print job; default value.

=NO

Only the following three attributes are taken into account:

- font (CHARACTER TYPE)
- language (LANGUAGE)
- NLQ (NEAR-LETTER-QUALITY)

## CHARS-POOL

Operand for output to laser printers; designates a character set pool with up to 64 character sets. Only HPFILEs may contain one or more character set pools; the spoolout job is (automatically) executed on an HP laser printer. Use of more than one character set requires the specification of CONTROL=PHYS; the name of the first character set and the number of specified character sets are displayed in the output for the command STATUS tsn.



The header line for each page (HEADER operand) is printed with the character set specified by "number".

When using HP laser printers please note:

The system administrator can specify whether the title page is to be printed with the standard character set or with the specified character set.

SHOW-SPOOL-PARAMETERS is used to interrogate the setting. The information is contained in the HEADER-PAGE and CHARACTER SET output fields.

See also page 376, "Loop/IM/BI/FORM records".

=(pool,number)

"pool" is the name of the character set pool. All character sets of this pool are loaded in the character set buffer for execution of the spoolout job.

"number" is the number of a character set from the character set pool;  $1 \leq \text{number} \leq 64$ . Default value: number = 1.

The number of the character set is determined by its position in the definition of the character set pool. The specified character set is used whenever CHARS-POOL is specified together with CONTROL=NO.

CHKPT

Only for output to the LP65 printer. Specifies that the print is to be controlled by the printer controller (for improved error handling; for details refer to "SPOOL System" manual [19]).

=ONPAGES

Specifies page-based print control.

=SECTION

Specifies area-based print control.

CONTROL

### 1.) Output to laser printer;

CONTROL defines whether control characters which are to be interpreted and which are specific to the laser printer are contained in your output file (see also the tables under the DEVICE operand and in the "SPOOL System" manual [19]).

=NO

Control characters within the text are not interpreted; default value  
For this reason no alternative character set is possible, i.e. the form is printed with the standard character set or with the first character set specified under CHARS, if one is specified.

=PHYS

Laser-printer-specific control characters in the file are interpreted. The first character in each record is interpreted as a feed control character. For ISAM files, the record key must appear at the beginning of each record.

Multiple character

sets can be used for printing out the file. The appropriate control characters may be located at any position in the text. For further details see also the "SPOOL System" manual [19].

With CONTROL=PHYS, the operands SHIFT, COPIES=(,number2) and LINES are ignored; SPACE=E is set implicitly.

CONTROL=PHYS is ignored if BINARY=YES is specified.

ENDNO should not be specified together with CONTROL=PHYS, since SPOOL does not take the number of control characters in a record into account.

## 2.) Output to RSO printers:

This operand controls the interpretation of character set identifiers, printer control characters, 9025 commands and VTSU codes, thus permitting the printing of files with several different character sets. (The record types mentioned below are described in the "RSO" manual.)

=NO

Specifies that type A-1 or A-2 records are to be printed, i.e. records containing print data only, except for perhaps one control character in the first byte.

Character set identifiers, printer control characters, 9025 commands and VTSU codes are not interpreted. A blank is substituted for the character "\". Non-printable characters, i.e. characters with the hexadecimal value < X'40' are output as blanks.

CONTROL=NO is the default value.



Specification of CONTROL=NO implies SPACE=1

=LOGICAL

Specifies that type B-1 or B-2 records are to be printed, i.e. records which, in addition to a feed control character in the first byte, can also contain data combined with

- character set identifiers
- printer control characters
- 9025 commands and
- VTSU codes.

With the exception of the VTSU codes VPA, NP, VT, NL and CR, which are output as blanks, all of the above-mentioned control characters are interpreted.

A character set identifier, a VTSU code or a 9025 command remains valid until a new control character is specified.

If a character set identifier is omitted, the standard character set for the form is printed by default.

Page feed specified with CONTROL=LOGICAL is performed using a LOOP record or constant line feed, for which reason the 9025 commands \LF, \FF and \CR are replaced by blanks. Setting of the form height is not permitted.



**=LINEMODE**

Specifies that records of type C are to be printed, i.e. records containing both data and LINEMODE control characters. LINEMODE control characters include (in any combination)

- printer control characters (i.e. physical control characters beginning with X'27' or X'3C'),
- 9025 commands and
- VTSU codes.

Control of the record and file formats (including page feed and line feed using LINEMODE control characters) is your responsibility. In the case of the 9025 Printer, you must also ensure that the initial paper positioning is correct.

A LOOP record will not be interpreted. Non-printable characters, i.e. characters with a hexadecimal value < X'40' are output as blanks.

When CONTROL=LINEMODE is specified, the spoolout job is executed using the standard character set of the form being used. unless the user changes the character set within the file with the aid of LINEMODE control characters.



Simultaneous specification of CONTROL=LINEMODE together with the ENDNO, HEADER, LINES, SHIFT, SPACE and STARTNO operands can lead to conflicts.

**=PHYSICAL**

Specifies that type D-1 or D-2 records are to be printed, i.e. records containing data mixed with LINEMODE control characters (see also CONTROL=LINEMODE). In contrast to CONTROL=LINEMODE, however, non-printable characters are also passed to the printer. You are responsible for the correctness of the control characters in the file (including feed control characters). If the SPACE operand is omitted, the default value SPACE=NO is assumed and you must implement page and line feeds by inserting LINEMODE control characters in the file.

Specification of SPACE=E for type D-2 records results in the feed control character in the first byte being interpreted as a line or page feed control character.



CONTROL=PHYSICAL must not be specified at the same time as any of the operands BINARY, CHARS-MODIFICATON=NO, DIA, FROM, HEADER, LINE or TO.

**=TRANSPARENT**

Specifies that type E-1 or E-2 records are to be printed, i.e. records containing any characters except for VTSU codes for RSO. Except for the feed control character in the first byte of type E-2 records, all characters of a set are passed to the printer without being interpreted by RSO.

You as user are responsible for the correctness of the control characters in the file (including feed control characters). If the SPACE operand is omitted, the default value SPACE=NO is assumed, and you must place your own LINEMODE control characters in the file to effect line feed or page feed.

Specification of SPACE=E for records of type E-2 results in the feed control character in the first byte being interpreted as a line or page feed.

If the SHIFT operand is specified at the same time, blanks are inserted at the beginning of each record; during printout, these blanks indent the text by the desired number of columns.



CONTROL=TRANSPARENT must not be specified at the same time as any of the operands BINARY, CHARS-MODIFICATION=NO, DIA, FROM, HEADER, LINE or TO.

**COPIES=number** **1) Output to printer, batch terminal or tape:** COPIES specifies how many additional copies of the file are to be printed. The number may be enclosed in parentheses. Each additional copy is given its own header page.

Possible values:  $0 \leq \text{number} \leq 255$ .

Default value: number=0, i.e. no additional copies are printed.

(number1,number2)

**2)Output to laser printers:** "number1" defines how many extra copies of the file are to be printed. Each additional printout is given its own header page.

Default value: number1 = 0, i.e. no additional copies.

"number" defines how often each individual page is to be printed in succession.

Default value: number2 = 0, i.e. no additional pages.

Possible values:  $0 \leq \text{number1}, \text{number2} \leq 255$ .

The COPIES operand must be used if multiple printouts are to be generated on a laser printer, since this device does not permit the use of copy paper. Consequently, existing user programs generating multiple printouts must be modified when the laser printer is used.



Each PRINT command with the operand COPIES=(,number2) is rejected if the LINES operand is specified at the same time and gives a number of lines which is greater than that in the LOOP record, minus the number of lines before the line with CHANNEL1.

A maximum of 255 copies of a page can be printed consecutively on an HP laser printer.

COPIES=(,255) has the same effect as COPIES=(,254), i.e. one original and 254 copies are printed.

- DEFER**                    Operand for output to printers, batch terminals or tapes; defines the conditions for output to batch terminal; valid for RBP printers only.
- =NO                    The receiver (see USER) operand must be active; default value.
- =YES                    The receiver (see the USER operand) must request output using the ROUT command (job name SOUTtsn) if no PNAME has been specified.  
                               The operand is ignored in the case of RSO and RBP printers.
- DESTINATION**            Operand for output to local or RSO printer; specifies a printer or pool of printers on which the spoolout job is to be executed.
- =device                Name of the RSO printer.  
                               The printer must have been defined using the ADD-SPOOL-DEVICE or EDIT-SPOOL-DEVICE parameter file (system administrator function).

=pool

Name of the device pool via which output is to be directed. A local pool contains a number of local devices and an RSO pool contains a number of RSO devices. These devices are entered in a list bearing the name of the pool. This list must be stored in the SPOOL parameter file. The list is managed by means of the ADD-PRINTERS-POOL and MODIFY-PRINTERS-POOL statements (by the system administrator or the RSO device manager).

Specifying DESTINATION=pool allows the print job to be output to any printer in the pool. The job is rejected if

- not at least one printer type is assigned to the form requested explicitly (FORM operand) or implicitly in the PRINT command (see also the SHOW-SPOOL-FORMS command in the "RSO" manual [20]), or
- one of the operands DIA or COPIES=(number1,number2) has been specified, both of which are only supported in RSO mode for 9025 Printers, and there is no 9025 Printer in the specified pool.

You should bear in mind that the pool can contain various types of devices. If the file to be printed contains control characters which are interpreted only by a certain type of printer, a printer of this type must be specified in the PRINT command. One way of doing this is by using the PRINT command to specify a form which is defined for the desired printer type only.

DEVICE

**1.) Output to printers, batch terminals or tapes:**

DEVICE designates the printer to which output is to be directed (see also page 375, "printer selection").

$$= \left\{ \begin{array}{l} \underline{\text{CENTRAL}} \\ *CENTRAL \end{array} \right\} (\text{DEVICE-TYPE} = \left\{ \begin{array}{l} *ALL \\ ND \\ HP \end{array} \right\})$$

Output to a local printer; default value. CENTRAL must not be abbreviated, whereas \*CENTRAL can be abbreviated from right to left, provided it remains unambiguous.

When selecting the printer type, the value of the CONTROL operand must be taken into account (see table below).

DEVICE-TYPE	CONTROL	
	=NO (default value)	=PHYS
=* <u>ALL</u>	All printers can handle the output.	Output can be handled on an ND laser printer (3350/3352) or an HP laser printer (3351/3353). With HP laser printers, however, the ND control characters are converted into HP control characters.
	The following also applies to HP printers: - The OVERPRINT function is converted to the LINE-MERGE function - The character X'FF' is converted to X'1F' (even if BINARY=YES is specified).	
=ND	The print job can only be executed on ND laser printers (3350/3352). The file may contain the character X'FF'. The OVERPRINT function is executed.	
=HP	Output can only be handled by HP laser printers (3351/3353). The OVERPRINT function is converted to the LINE-MERGE function.	
	Restrictions:  - HPFILE: The file must not contain the character X'FF'.  - NDFILE: X'FF' is converted to X'F1'.	Support of the HP-specific control characters:  Restriction: The file may contain the character X'FF' as an escape character only if the control character interpretation function has been activated (DATA MODIFICATION=NO)

*Note for HP laser printers:*

If the entries in the PRINT command specify output to an HP laser printer, an HPFILE must be provided, otherwise the command is rejected.

If no HP laser printer is provided in the configuration or if the available HP laser printers are inadequately configured (e.g. insufficient fonts, no graphics memory for FOBs), the jobs can only be output on replay tapes.

The system administrator can request information on such jobs by issuing the STATUS command.

**=REMOTE** Output is directed to a printer at a batch terminal. The spoolout job is placed in the appropriate queue. Conditions for output and reception may be specified using the operands DEFER, USER and T. If DEVICE=REMOTE is specified, up to 132 characters per line can be printed.

**2.) Output to RSO printers:**


DEVICE specifies an RSO printer to which output is to be directed. DEVICE must not be specified at the same time as the DESTINATION operand.

**=device** Name of the RSO printer. The printer must be defined (by the system administrator) with the ADD-SPOOL-DEVICE or EDIT-SPOOL-DEVICE command in the SPOOL parameter file. The DEFER, USER, T, TAPE, VOLUME and RETPD operands must not be specified at the same time as DEVICE=device.


**DEVIN=TAPE** Operand for output to printers, batch terminals or tapes; specifies that the file to be output is a SAM file on tape. Foreign tape files without labels or with non-standard labels cannot be output.

The following must be noted when specifying the DEVIN operand:

- The DEVIN operand must not be specified together with TAPE=YES/T9P/T9G, i.e. a tape file cannot be output to another tape.
- In the case of foreign files on multifile tapes, only the first file is output.
- If LOCK=YES is specified at the same time as DEVIN=TAPE, the spoolout job will be rejected.

- DIA**                    Operand for output to laser printers; designates a forms overlay for printing.  
If this operand is omitted, no forms overlay is used.
- =cc**                    Designates the forms overlay (2 alphanumeric characters), which must first be confirmed by the system administrator.
- DACCOUNT=daccount**                    Indicates the account number for the user ID specified with **DUSER**.
- DPASSWD=dpasswd**                    Indicates the password for the user ID specified with **DUSER**.
- DUSER=duser**                    Indicates the user ID under which the print job is to be executed.
- ENDNO**                    Operand for output to printers, batch terminals or tapes; permits specification of a byte number (record column) up to which the records are printed. (The bytes of a record are numbered consecutively from left to right, beginning with 1).  
Specification or omission of the **SPACE=E/A/I** operand has the following effects:
- Specification of **SPACE=E/A/I**:  
Output terminates with the data byte following the specified byte number. (Exception: specification of **STARTNO=0** for an ISAM file with **KEYPOS=5** terminates output with the data byte corresponding to the specified byte number.)
  - Omission of **SPACE=E/A/I**:  
Output terminates with the data byte corresponding to the specified byte number. If the records are longer than the formal definition permits, they will be continued in the next line (continuation line).
-  **ENDNO** should not be specified together with **CONTROL=PHYS**, since **SPOOL** does not take the number of control characters in a record into account.

=byteno            Number of the data byte;  $1 \leq \text{byteno} \leq 32767$ .  
Default value:  
– End of print line (136).  
– byteno = 2048 for an EAM file, SYSLST or SYSOUT.

 If ENDNO is specified together with CCPOS and if "ccpos" is < "byteno", the character at the position "byteno+1" is also output since the feed control character declared with CCPOS is not taken into account for ENDNO and is not output.

FAMILY            Operand for output to printer, or tape; defines whether a common TSN is to be assigned when a number of files or library members are specified in the PRINT command.

=STD               Specifies that the defaults from the SPOOL parameter file are to be used for spoolout jobs on local printers. You can display this value on the screen by using the SHOW-SPOOL-PARAMETERS command. The default values may be different for RSO printers and local printers.

=YES               Only for local spoolout jobs; specifies that a common TSN is to be assigned. Parallel processing is not possible.

=NO                Each file and library member is to be printed with its own TSN. Several print jobs can thus be processed in parallel.

FOB                Operand for output to laser printers; designates a forms overlay buffer (FOB) for superimposing (on the print page) text and graphics which are not contained in the file to be printed. When the FOB operand is specified, the file is (automatically) output to the HP laser printer with graphics memory. For further details see the "SPOOL System" manual [19].  
Use of an FOB for the spoolout job is recorded in the output of the command STATUS tsn.

=fob               Name of the FOB# containing the text/graphics to be used.



**FORM** Operand for output to printers, batch terminals or tapes; designates the paper (type of form) to be used for output (e.g. STD, STDSF1, STDWA4).

STD: 12-inch \* 315 mm recycling paper

STDSF1: 12-inch \* 375 mm recycling paper for chain printers.

STDWA4: 12-inch \* 240 mm white paper, DIN A4, with perforations.

The name must be defined in the SPOOL parameter file and can be interrogated by means of the SHOW-SPOOL-FORMS command. The SPOOL parameter file also includes a specification as to whether or not a header or trailer page is to be printed.

=STD

Standard form; default value.

=form

Designates the form type (type of paper).

"form" can only be specified for printers with a vertical format buffer (VFB).

A LOOP record is assigned to the form type; this LOOP record must be contained in a certain character set file, determined by the printer type.

The following table indicates which character set file must contain this LOOP record and for which printer type:

Output device	File with the LOOP record for the specified form type
3337, 3338, 3339, 3348, 3365 Printer	\$TSOS.SPOOLFILE
3350, 3352 Laser Printer	\$TSOS.NDFILE or the file specified in the IMAGE operand
3351, 3353 Laser Printer	\$TSOS.HPFILE or the file specified in the IMAGE operand

The LOOP record assigned to the specified form type is ignored if the LOOP operand is specified at the same time.

If no FORM and LOOP operands are specified, the standard form is used by default.

A LOOP record explicitly specified in the LOOP operand must have the same length as the LOOP record specified for the form being used.

See also page 375, "Printer selection".

FROM	Operand for output to printers, batch terminals or tapes; permits the specification of a page (P) or line (L) of the print file with which output is to begin. The pages of the print file are specified as defined in the LINES operand unless a control character initiates an earlier page break; see also Example 3.
=[-]mP	Output begins at page m; $1 \leq m < 1073741823$ . P = default value.
=[-]mL	Output begins with line m; $1 \leq m < 1073741823$ . "-m" means that only the last m pages or lines of the file are to be output. If the number specified by "-m" is greater than the total number of lines or pages in the file, the entire file is output.
HDRNUM=n1	Operand for output to printers, batch terminals or tapes; selects the different header pages. Possible values: $0 \leq n1 \leq 2^{31} - 1$ The exact meaning of the operand depends on the specific computer center definitions.
HEADER	Operand for output to printers, batch terminals or tapes; specifies that each page (with the exception of the header and trailer pages) is to contain a header line. The entries D, T, P and S define the format and text in the header line.

=NO A header line is not printed; default value.

=[D],[T],[P]

The header line consists of 3 text sections. One of the entries D, T or P is assigned to each of these sections, as shown in the following table:

HEADER	Format of the header line <sup>1)</sup>		
	As of column 1	As of col.21	As of col. 124 (77)
=D	DATE yy-mm-dd	——	——
=T	——	First record	——
=P	——	——	PAGE nnnn
=(D,T,P)	DATE yy-mm-dd	First record	PAGE nnnn

1) The column specification enclosed in parentheses applies to form definitions with a line length ≤ 80 characters.

**where:**

yy-mm-dd                    Year-month-day  
 First record                First record in the file  
 nnnn                         Page number

=S The header line has the following format:

HEADER	Format of the header line <sup>1)</sup>			
	As of column 1	Column 41 (11)	Column 60 (21)	Column 124 (77)
=S	DATE yy-mm-dd	userid	filename	PAGE nnnn

1) The column specifications enclosed in parentheses apply to form definitions with a line length ≤ 80 characters.

**where:**

yy-mm-dd                    Year-month-day  
 userid                      User identification  
 filename                    File name  
 nnnn                         Page number

The values for "column" apply only if the SHIFT operand is omitted. If SHIFT is specified, the values specified therein apply, but the header line is truncated after column 132.

Unless otherwise specified, the header line is followed by a blank line.

When HEADER=...,T,... is specified, the first record of the file is not repeated later.

IMAGE

Operand for output to laser printers;

Designates a user file which may contain LOOP records (see FORM or LOOP operand), character sets (see CHARS operand), FOB records and CHARS-POOL records (see CHARS operand) (parameter file).

If this operand is omitted, the relevant information is obtained from the \$TSOS.NDFILE, \$TSOS.HPFILE or \$TSOS.RSOFIL file.

=image

Part of the file name "image.NDFILE", "image.HPFILE" or "image.RSOFIL". "image" may include a catalog ID and a user ID and must not comprise more than 28 alphanumeric characters plus the catalog ID and the user ID.

If the image.NDFILE, image.HPFILE or image.RSOFIL file does not exist under the ID of the user submitting the job, the \$TSOS.image.NDFILE, \$TSOS.image.HPFILE or \$TSOS.image.RSOFIL file is used.

If "image" contains a user ID and/or a catalog ID, the operating system only searches under this/these ID(s).

If you require output with page rotation and are working with your own "image.HPFILE" file, this HPFILE must contain the information on FOBs and character sets or character pools, otherwise, the \$TSOS.image.HPFILE is searched for this information (see also page 374 "Use of character sets").

LINES=lines

Operand for output to printers, batch terminals or tapes; specifies how many lines (including the header line and blank lines) are to be printed on one page.

If no specification is made for LINES, the number of lines per print page is calculated in accordance with the following formula, depending on the specification in the HEADER operand:

$$\text{Number of lines} = P * Z - A - 6$$

**where:**

P = paper size in inches

Z = line density

A = number of lines before the first channel 1

The LINES operand is ignored if CONTROL=PHYSICAL has been specified.

**Printers with a loadable vertical format buffer:**

- Channel 1 controls the line at which printing starts. It is standard for 2 blank lines to appear before the first printed line, i.e. in the LOOP record, channel 1 (CHANNEL 01) is in the third line.
- If "lines" is greater than the number of lines specified in the LOOP record, the value specified in the LOOP record is used.
- "lines" must be at least three times as large as the number of lines specified in the SPACE=1/2/3 operand if the LINES operand is specified together with the HEADER and SPACE operands.

LOCK  
=YES

Operand for output to printers, batch terminals or tapes; locks the file against updates for as long as the spoolout job is in the wait state (TYPE 4, see the output from the "STATUS LIST" command) or if it is aborted (see the CANCEL command). During this time the file can be read only.

File protection via LOCK=YES remains in effect even if the spoolout job does not begin until the next session. From the beginning to the end (or abortion) of the spoolout job the file automatically remains locked.

LOCK=YES is ignored if one of the operands \*, \*SYSLST or \*SYSOUT is specified or if it concerns a PLAM library.

Files on tapes are never locked. If DEVIN=TAPE is specified together with LOCK=YES, the job will be rejected.

A spoolout job will be created even if the file to be output has been reserved using a SECURE command. However, the reservation must have been lifted by the time the spoolout job is running, otherwise the job cannot be executed.

The file to be output will remain locked until the end of the system run if the operand LOCK=YES has also been specified in the PRINT command and the job could not be executed due to this reservation.

=NO

The file is not locked while the spoolout job is waiting in queue. The file can be deleted or updated until the spoolout job has actually begun.

The default value for the LOCK operand is defined in the SPOOL parameter file. LOCK=NO is ignored in the case of temporary files.

LOOP  
=loop1

**1.) Output to printers, batch terminals or tapes:**

name of the LOOP record (which is to be loaded into the vertical format buffer of the printer. ("loop1" consists of up to three alphanumeric characters, except for "@", "\$" and "&".)  
The length of the specified LOOP record must be the same as the length of the standard LOOP record for the form used.

This specification is required for output to a laser printer or to a 3337, 3338, 3339, 3348 or 3365 printer.

LOOP records are part of the character set file (NDFILE, HPFILE, ....). The following assignments for character set file and printer are set by default:

File name	Output device
\$TSOS.LP37FILE	3337, 3338, 3339 3348, 3365 Printer
\$TSOS.NDFILE	3350, 3352 Laser Printer
\$TSOS.HPFILE	3351, 3353 Laser Printer

If no LOOP record is specified, the implicit specifications in the FORM operand are used. If neither FORM nor LOOP is specified, default values are assumed.

LOOP

**2.) Output to laser printers:**

LOOP designates one or two LOOP records to be loaded into the vertical format buffer for feed control purposes. The second LOOP record ("loop2") is required if page rotation through 90°/270° is specified (see the ROTATION... operand).

If the "loop1/loop2" specification is omitted, the LOOP records implicitly specified in the FORM=... operand are assumed. The SHOW-SPOOL-FORMS command can be used to inquire whether a standard LOOP record is provided for the paper (type of form) to be used.

- =loop1            Name of the LOOP record for output to ND/HP laser printer.
- =(loop1,loop2)    "loop1" and "loop2" specify LOOP records for output of a file with some pages rotated through 90<sup>0</sup>/270<sup>0</sup>. "loop1" and "loop2" are only interpreted when output is to an HP laser printer.
- "loop1" is the name of the LOOP record used to format the pages which are not to be rotated.  
                  "loop2" is the name of the LOOP record used to format the pages which are to be rotated.
- =(,loop2)        Name of the LOOP record used to format the pages which are to be rotated.
- PAGECC            Specifies whether the file contains page control characters which are to be interpreted.
- =STD            The specifications in the print parameter file apply.
- =NO             The control characters are not to be interpreted.
- PNAME=pname     Operand for output to printers, batch terminals or tapes; job name for the spoolout job.  
                  "pname" can consist of up to 8 characters from the set (A,...,Z, 0,...,9, @, #, \$, ., -), but must not begin with a hyphen or end with a period.  
                  "pname" may begin with a period only if this is followed by an alphabetic character; the period itself, when part of "pname", is not printed out on the header page in this case.
- The job name is printed on the header page in the third oversize-letters line and also appears in the output of the STATUS command. When the specification is omitted, the job name from the LOGON command is used in its place.
- Outputs to a remote batch printer:** The "pname" field only appears in the first print job. With all other print jobs, this field is empty.

<p>PRINTER =136</p>	<p>This operand is no longer interpreted by SPOOL.</p>
<p>PRIORITY  =p</p>	<p>Operand for output to printers, batch terminals or tapes; defines the start priority of the spoolout job relative to other spoolout jobs.</p> <p>Priority of the spoolout job. <math>\text{MAXIMUM} \leq p \leq 255</math>. The value for MAXIMUM is defined in the JOIN entry and can be interrogated by means of the SHOW-USER-ATTRIBUTES command.</p> <p>If impermissible values are specified for "p" or if the PRI operand has been omitted, the operating system uses the priority of the generating job.</p>
<p>RESOURCE=LP65(...)</p>	<p>Specifies the resources to be used for print jobs on the LP65 printer (see "SPOOL System" manual [19]).</p>
<p>PAGEDEF=number</p>	<p>Specifies the number of the PCL (Printer Control Language) to be used.</p>
<p>DIAV=diav</p>	<p>Specifies the forms overlay to be used for the front of the page.</p>
<p>DIAR=diar</p>	<p>Specifies the forms overlay to be used for the reverse side of the page.</p>
<p>INTRAY=nnn</p>	<p>Specifies the IN tray.</p>
<p>OUTRAY=nnn</p>	<p>Specifies the OUT tray.</p>
<p>DUPLEX =STD =NO =YES =TUMBLE</p>	<p>Specifies single-side or double-side printing.</p> <p>The DUPLEX value defined in the PCL is to be used.</p> <p>Specifies single-side printing.</p> <p>Specifies double-side printing.</p> <p>Both sides are to be printed, the pages being turned at the narrow margin and not, as is customary, at the wide margin.</p>
<p>RETPD  =days</p>	<p>Operand for output to printers, batch terminals or tapes; defines a retention period (in days) for the output file, i.e. the file cannot be updated or deleted during this time, but only read.</p> <p>Retention period in days (0,...,999). Default value: days=0, i.e. the file can be updated immediately.</p> <p>If the retention period (RETPD) of a file to be written to a tape specified by VOLUME is greater than the retention period specified</p>



for the last file written to this tape, the file is written to a scratch tape rather than to the specified tape.

## ROTATION

Operand for output to printers, batch terminals or tapes; permits rotation of the page. The print page formatted in the printer is rotated (clockwise) through a certain number of degrees and printed on the paper; for example, paper inserted vertically can also be printed in landscape mode.

A special LOOP record is required for page rotation through  $90^0/270^0$  (see the LOOP operand). Output is (automatically) to HP laser printer unless ROTATION=NO is specified.

The SHOW-SPOOL-PARAMETERS command indicates whether or not page rotation is possible; output field DEVICE-TYPE:..., ROT=YES/NO.

Spoolout jobs using page rotation are displayed in the output of the commands STATUS LIST and STATUS tsn.

=NO

Page rotation is not carried out. Any control characters specifying page rotation in the file are not interpreted, even if CONTROL=PHYS is specified; NO is the default value.

=YES

Page rotation is carried out; simultaneous specification of CONTROL=PHYS is required. (The file to be printed must contain the control characters for page rotation; see the "SPOOL System" manual [19].)

=90  
=180  
=270

}

Each print page is rotated (clockwise) through  $90^0/180^0/270^0$  and printed on the paper. Control characters for page rotation in the file are not interpreted. A special LOOP record is required for page rotation through  $90^0/270^0$  (see LOOP operand).

=(0,180)  
=(90,270)  
=(180,0)  
=(270,90)

}

The numbers enclosed in parentheses specify the number of degrees through which a page is rotated. The first number refers to odd-numbered print pages (1st, 3rd, 5th and so forth), the second to even-numbered pages (2nd, 4th, 6th, etc.). Any control characters for page rotation in the file are not interpreted.

A separate LOOP record is required for rotation through  $90^0/270^0$  (see LOOP operand).

- SECTION=(...)** Restricts print output to one particular area of the file. In order to mark this area, the user defines records containing a character string to be specified at a certain position; this position must be specified precisely.
- ID=string** Specifies whether or not the area with records is marked. "string" specifies the character string, with the records marking the area. "string" may be alphanumeric or hexadecimal
- POS=n** Specifies the position of "string" within the record.
- FIRST=first** Specifies where the area to be output begins. "first" may be an integer, an alphanumeric character string or a hexadecimal character string.
- POS=n** If "first" is a character string, "n" specifies at which position within the record the character string is located.
- NUMBER=** Specifies the record with the specified character string at the specified position where output is to begin; default value: 1.
- LAST=last** Specifies where the area to be output ends. "last" may be an integer, an alphanumeric character string or a hexadecimal character string. The record marked "last" is not output.
- POS=n** If "last" is a character string, "n" specifies at which position within the record the character string is located.
- NUMBER=** Specifies the record with the specified character string at the specified position where output is to end.
- SHIFT=columns** Operand for output to printers, batch terminals or tapes; for all printers, with the exception of RSO printers: indentation of the output text by the specified number of columns. ("columns" = {0,1,...,31}). The default value for "columns" can be interrogated using the SHOW-SPOOL-PARAMETERS command; the information is taken from the output field PRINT-CMD-DEFAULTS:...SHIFT=.... The SHIFT operand is ignored if CONTROL=PHYS is specified at the same time.

## SPACE

**1.) Output to printers, batch terminals or tapes:**

SPACE defines the number of line feeds or the type of feed control characters contained in the file to be output.

=1 }  
 =2 }  
 =3 }

Number of line feeds after printing a line;  
 Default value: SPACE=1.

=E  
 =A  
 =I

The file contains SIEMENS feed control characters.

The file contains ASA feed control characters.

The file contains IBM feed control characters.

For more details on feed control characters see page 376.

default value: SPACE=E for an EAM file, SYSLST and SYSOUT.

If SPACE=E is specified in the PRINT command for files created with RECFORM=(...,A), SPOOL automatically assumes SPACE=A.

The following points must be noted when the SPACE operand is specified:

- Specification of CONTROL=PHYSICAL implies SPACE=E.
- Files created with RECFORM=U cannot be output with SPACE=E.
- In the case of older chain printers (e.g. the 3340), a feed to channels 9 and 12 is not possible.

**2.) Output to RSO printers:**

SPACE defines the number of line feeds or the type of feed control characters contained in the file to be output.

=1 }  
 =2 }  
 =3 }

Number of line feeds after printing a line;  
 default value: SPACE=1.

=NO

Specifies that no line feed is intended for the file.

You as user are responsible for line feed, i.e. each page or line feed must be effected by inserting appropriate printer control characters in the file.

For this reason the operand value NO is supported only if CONTROL=PHYSICAL or CONTROL=TRANSPARENT has been specified. The PRINT command is rejected if neither

CONTROL=PHYSICAL nor CONTROL=TRANSPARENT has been specified in conjunction with SPACE=NO.

- =E The file contains SIEMENS feed control characters.
- =A The file contains ASA feed control characters.
- =I The file contains IBM feed control characters.

If SPACE=E/A/I has been specified in a print job, the feed control characters are interpreted based on the value specified in the CONTROL operand. When CONTROL=PHYSICAL or CONTROL=TRANSPARENT is specified, note that line and page feeds cannot be controlled via LOOP records. The following feed control characters would then result in an error:

- X'C2',X'C3',...,X'CB'
- X'82',X'83',...,X'8B'

## STARTNO

Operand for output to printers, batch terminals or tapes; permits specification of a byte number (record column), starting with which the records of a file are to be output. (The bytes of a record are consecutively numbered from left to right, beginning with 1.)

Specification or omission of the SPACE=E/A/I operand has the following effects:

- Specification of SPACE=E/A/I:  
Output begins with the data byte following the specified byte number. The feed control character is interpreted independently of the STARTNO=... entry, provided that the value of START is smaller than the length of the record. If the specified value is greater than or equal to the length of the record, the record is skipped, i.e. there is no printing and no feed.
- Omission of SPACE=E/A/I:  
Output begins with the data byte corresponding to the specified byte number.

## =byteno

Number of the data byte;  $0 \leq \text{byteno} \leq 32767$ .

With byteno = 0 the following points must be noted:

- Control character in the first data byte (SAM file or ISAM file with KEYPOS  $\geq 6$ ): output is as for STARTNO=1.
- Control character not in the first data byte (ISAM file with KEYPOS = 5): output begins with the first data byte, regardless of whether or not SPACE=E/A/I has been specified.

Default value: byteno = 1.



If "byteno" is greater than the length of the record, the record will not be printed. ISAM master key and control characters are part of a record.

If STARTNO is specified together with CCPOS and "ccpos" is < "byteno", the first character output is the character at position "byteno+1" since the feed control character declared with CCPOS is not taken into account when the position for STARTNO is calculated.

T=staid

Operand for output to printers, batch terminals or tapes; defines the batch terminal to which output is directed. "staid" is the terminal identifier (see the RSTART command). The PRINT command is rejected if an invalid station ID is specified.

TAPE

Operand for output to printers, batch terminals or tapes; determines whether the file is to be output to printer or tape and permits the specification of a specific recording density for output to tape.

The following points must be borne in mind in conjunction with output to tape:

- TAPE=YES/T9P/T9G... must not be specified together with the DEVIN operand.
- Files on disks to be output to tape (TAPE=YES/T9P/T9G...) can be locked by specifying LOCK=YES; files on tape, however, cannot be locked.
- If the retention period (RETPD) of a file to be written to a tape specified by VOLUME is greater than the retention period of the last file written to this tape, the file is written to a scratch tape rather than to the specified tape.
- The name of an output file created via the TAPE operand may be up to 54 characters in length. This must be taken into account when the input file name is assigned.

=NO

Outputs the file to printer; default value.

=YES

Outputs the file to tape, using the highest available recording density.

A SAM tape file with standard labels is created as the output file. It is written to tape under the user ID of the calling job and is given the following name:

$$\left. \begin{array}{l} \{ S.tsn. \} \\ \{ S.pname. \} \end{array} \right\} \left\{ \begin{array}{l} [hhmmss.] filename \\ [EAMno] \end{array} \right\}$$

- tsn            Task sequence number of the spoolout job.
- hhmmss       Time of day if several files are output with the same job name and the same file name (hh = hours, mm = minutes, ss = seconds).
- filename      Name of the cataloged input file.
- pname         Job name if the PNAME operand has been specified.
- EAMno        EAM file number.

SPOOL creates a multifile tape set if this operand is used to output any of the following:

- more than one file to the same tape
- file generations
- more than one file designated by a partially qualified file name

The tape file can be output to printer using the PRINT command (DEVIN=TAPE operand). A tape file cannot be output to another tape.

```
=T9P
=T1600
=T9G
=T6250
=TAPE-C1
```

Specifies the recording density with which the file is to be output to tape. The entries T9P and T1600, as well as T9G and T6250, are equivalent entries.

TEXT='text'

Operand for output to printers, batch terminals or tapes; the information input here (max. 32 characters) is stored in the SCB for use in processing system exits.

The first 8 characters are printed in oversize letters on the header page under "Mailing Box".

TO

Operand for output to printers, batch terminals or tapes; permits specification of a page (P) or line (L) in the print file, with which output is to be terminated. The pages of the file are printed as defined in the LINES operand (unless a control character initiates an earlier page break); see also Example 3.

=n[P]

Output ends with page n;  $1 \leq n < 1073741824$ .

=n[L]

Output ends with line n;  $1 \leq n < 1073741824$ .



If the operand is specified in the form "FROM=m, TO=n", the

following applies:

- TO can only be specified as TO=n.  
P or L is taken from FROM=...
- $m \leq n$ .
- TO cannot be specified further if the entry for FROM is a negative number.

## TRANSLATION-TABLE

Operand for output to RSO printers; specifies a code translation table which is activated for the duration of the print job.

`=(name1)` Specifies the name of a translation table which has been defined in the file \$TSOS.RSOFIL. "name1" consists of up to 8 alphanumeric characters.

`=(name1,name2)` Specifies the name of a translation table defined in the private user file \$userid.name2.RSOFIL. "name1" consists of up to 8 alphanumeric characters. "name2" consists of either a maximum of 4 alphanumeric characters starting a letter, or has the same format as the specification for the IMAGE operand (page 412)

**TRAY** Operand for output to RSO printers; specifies the number of the tray from which the print paper is to be taken.

`=number` Specifies the number of the tray from which the print paper is to be taken.  
The paper defined under "number" is used to print both the file and the header (HEADER-PAGE) and trailer (TRAILER-PAGE) pages. If the TRAY operand is omitted, the paper is taken from the tray specified as the default value in the SPOOL parameter file.  
Possible values for "number": 1...9.  
RSO does not check the specified values. The maximum number of trays for the various printer types is given in the "RSO" manual [20].

`TRLNUM=n2` Operand for output to printers, batch terminals or tapes; selects the different trailer pages.

Possible values:  $0 \leq n2 \leq 2^{31}-1$

The exact meaning of the operand depends on the respective computer center specifications.

TRUNC	Operand for output to printers, batch terminals or tapes; defines behavior in cases where lines are truncated.
=STD	The default from the SPOOL parameter file applies. This value can be displayed on the screen with the SHOW-SPOOL-PARAMETERS command (field ERROR-PR=(TRUNC=...)).
=IGNORE	Processing of the SPOOLOUT job is continued (i.e. ERASE=YES is ignored), and an appropriate warning is printed on the trailer page.
=KEEP	Processing of the SPOOLOUT job is terminated.
USER=userid2	<p>Operand for output to printers, batch terminals or tapes; indicates a user ID which is also to be authorized to receive the output, in addition to the user's own ID. "userid2" must also be specified in the RLOGON command at the batch terminal.</p> <p>Output must be requested by means of the ROUT command. Only the first receiver (own user ID or "userid2") requesting output will receive it. In the case of RSO, the operand will be ignored.</p>
VOLUME	<p>Operand for output to printers, batch terminals or tapes; permits specification of a volume serial number for output to magnetic tape.</p> <p>The following points must be borne in mind when specifying the VOLUME operand:</p> <ul style="list-style-type: none"> <li>– If the retention period (RETPD) of a file to be written to a tape specified by VOLUME is greater than the retention period of the file last written to this tape, the file is written to a scratch tape rather than to the specified tape.</li> <li>– A print job cannot process more than 256 files. If more than 256 files are specified in a PRINT command, several jobs are created. To prevent tapes from being overwritten, the VSN specifications are only used for the first job. Any other jobs work with scratch tapes.</li> </ul>
= <u>SCRATCH</u>	These scratch tapes are provided by the operator. If the operand TAPE=YES/T9P/T9G has been omitted, TAPE=YES is assumed by default.
=vsn =(vsn,...)	<p>Volume serial number of the tape to which output is to be directed. Up to 4 VSNs may be specified.</p> <p>The tapes are mounted in the specified order, using only the number of tapes actually required.</p> <p>If a specified tape cannot be accessed, SPOOL automatically</p>



provides a scratch tape.

If the operand TAPE=YES/T9P/T9G has been omitted, TAPE=YES is assumed by default.

### Incompatible operands in the PRINT command

Operand	Incompatible operands: A print job is rejected if at least one of the operands in the right-hand column is specified at the same time as the operand in the left-hand column.
BINARY=YES	SPACE=E/A/I/NO, CONTROL=LOG/PHYS/LINEMODE/TRANSPARENT, START-SPOOL=NO, CCPOS
CCPOS	SPACE=1/2/3, CONTROL=PHYS, HEADER=S/(T,D,P), LINES, USER, T, DEFER, DEVICE=REMOTE, BINARY=YES, START-SPOOL=NO, PAGECC=NO
CHARS	CHARS-POOL, DEVICE=REMOTE, T, USER, DEFER, PRINTER=136, START-SPOOL=NO, RESOURCE=LP65
CHARS-MODIFICATION	DEVICE=CENTRAL/CENTRAL (ND) /CENTRAL (HP) /REMOTE, START-SPOOL=NO, CONTROL=NO/TRANSPARENT, T, USER, DEFER
CHARS-POOL	DEVICE=CENTRAL (ND) /REMOTE/RSO-STATION, T, USER, DEFER, CONTROL=LOG/LINEMODE, PRINTER=136, START-SPOOL=NO, CHARS, TAPE, VOLUME, RETPD, RESOURCE=LP65
CONTROL=LINEMODE	SPACE=E/A/I, BINARY=YES, HEADER=S/(T,D,P), LINES, DEVICE=CENTRAL/CENTRAL (ND) /CENTRAL (HP) /REMOTE, T, USER, DEFER, DIA, CONTROL=LOG/NO/PHYS/TRANSPARENT, PRINTER=136, START-SPOOL=NO, LOOP=(,loop2), FOB, ROTATION=n/YES, CHARS-POOL, RESOURCE=LP65, PAGECC=NO
CONTROL=LOG	BINARY=YES, HEADER=S/(T,D,P), DEVICE=CENTRAL/CENTRAL (ND), DEVICE=CENTRAL (HP) /REMOTE, T, USER, DEFER, CONTROL=PHYS, CONTROL=LINEMODE/NO/TRANSPARENT, PRINTER=136, RESOURCE=LP65, START-SPOOL=NO, LOOP=(,loop2), CHARS-POOL, FOB, SPACE=NO, ROTATION=n/YES, PAGECC=NO
CONTROL=NO	CONTROL=LINEMODE/LOG/PHYS/TRANSPARENT, SPACE=NO
CONTROL=PHYS	SPACE=1/2/3, BINARY=YES, HEADER=S/(T,D,P), DEVICE=REMOTE, T, USER, DEFER, CONTROL=LOG/LINEMODE/NO/TRANSPARENT, PRINTER=136, START-SPOOL=NO, RESOURCE=LP65S, CCPOS
CONTROL=TRANSPARENT	BINARY=YES, HEADER=YES, HEADER=(T,D,P), LINES, DIA, CONTROL=LINEMODE/LOG/PHYS/NO, CHARS-MODIFICATION, T, USER, RESOURCE=LP65, DEVICE=CENTRAL/CENTRAL (ND) /CENTRAL (HP) /REMOTE
COPIES=n1	COPIES=(,n2), START-SPOOL=NO

COPIES=(,n2)	COPIES=n1, DEVICE=REMOTE, T, USER, DEFER, PRINTER=136, START-SPOOL=NO
DEFER	FROM=NP/L, FROM=-NP/L, TO=N/NP/L, DEVICE=CENTRAL/CENTRAL(ND), DEVICE=CENTRAL(HP)/RSO-STATION, CHARS, DIA, IMAGE, SECTION CONTROL=LOG/PHYS/LINEMODE/TRANSPARENT, COPIES=(,n2), CCPOS, LOOP=(,loop2), PRINTER=136, START-SPOOL=NO, CHARS-POOL, FOB, ROTATION=n/YES, RESOURCE=LP65, *LIB, LOOP, PAGECC=NO
DELETE-FILE=DESTROY	EAM file number, *, ERASE, DELETE-FILE=ERASE, DESTROY, START-SPOOL=NO/n
DELETE-FILE=ERASE	ERASE, DESTROY, DELETE-FILE=DESTROY, START-SPOOL=NO,
DIA	DEVICE=REMOTE, T, USER, DEFER, CONTROL=LINEMODE/PHYS/TRANSPARENT, PRINTER=136, START-SPOOL=NO
DESTINATION=device	DEVICE, DESTINATION=pool, T, USER, DEFER, PRINTER=136, FOB START-SPOOL=NO, LOOP=(,loop2), CHARS-POOL, ROTATION, RESOURCE=LP65, PAGECC=NO
DESTINATION=pool	DEVICE, DESTINATION=device, T, USER, DEFER, TAPE, VOLUME RETPD
DESTROY	EAM file number, *, ERASE, DELETE-FILE=ERASE, DELETE-FILE=DESTROY, START-SPOOL=n/NO
DEVICE=CENTRAL	DEVICE=CENTRAL(ND)/CENTRAL(HP)/REMOTE/RSO-STATION, T, USER, DEFER, CONTROL=LOG/LINEMODE/TRANSPARENT, START-SPOOL=NO, DESTINATION, TRANSLATION-TABLE, CHARS-MODIFICATION, TRAY, SPACE=NO
DEVICE=CENTRAL(ND)	DEVICE=CENTRAL/CENTRAL(HP)/REMOTE/RSO-STATION, T, USER, DEFER, TAPE, VOLUME, RETPD, CONTROL=LOG, CHARS-MODIFICATION, CONTROL=LINEMODE, START-SPOOL=NO, LOOP=(,loop2), CHARS-POOL, RESOURCE=LP65, FOB, ROTATION=n/YES, TRAY, TRANSLATION-TABLE, SPACE=NO
DEVICE=CENTRAL(HP)	DEVICE=CENTRAL/CENTRAL(ND)/REMOTE/RSO-STATION, T, USER, DEFER, TAPE, VOLUME, RETPD, CONTROL=LOG/LINEMODE/TRANSPARENT, START-SPOOL=NO, TRAY, CHARS-MODIFICATION, RESOURCE=LP65, TRANSLATION-TABLE, SPACE=NO

DEVICE=REMOTE	FROM=NP/L, FROM=-NP/L, TO=N, TO=NP/L, DEVICE=CENTRAL/ CENTRAL(ND)/CENTRAL(HP)/RSO-STATION, CHARS-POOL, DIA, IMAGE, CONTROL=LOG/PHYS/LINEMODE/TRANSPARENT, COPIES=(,N2), LOOP, LOOP=(,loop2), PRINTER=136, START-SPOOL=NO, CHARS, T, TRAY, ROTATION=n/YES, CHARS-MODIFICATION, DESTINATION, *LIB, FOB, TRANSLATION-TABLE, SECTION, CCPOS, RESOURCE=LP65, PAGECC=NO, SPACE=NO
DEVICE= RSO-STATION	DEVICE=CENTRAL/CENTRAL(ND)/CENTRAL(HP)/REMOTE, T, USER, DEFER, RESOURCE=LP65, PRINTER=136, START-SPOOL=NO, PAGECC=NO, LOOP=(,loop2), FOB, CHARS-POOL, ROTATION=n/YES, DESTINATION
DEVIN=TAPE	EAM file number, *, TAPE, VOLUME, RETPD, START-SPOOL=n/NO, LOCK=YES
EAM file number	Cataloged file name, *, *SYSOUT, *SYSLST, DESTROY, *LIB DELETE-FILE=DESTROY, DEVIN=TAPE, START-SPOOL=n/CLOSE/NO
ENDNO	START-SPOOL=NO
ERASE	DELETE-FILE=ERASE, DESTROY, DELETE-FILE=DESTROY, START-SPOOL=NO
FAMILY=YES	DEVICE=REMOTE, T, USER, DEFER, SECTION
FOB	DEVICE=CENTRAL(ND)/REMOTE/RSO-STATION, T, USER, DEFER, CONTROL=LOG/LINEMODE/TRANSPARENT, PRINTER=136, START-SPOOL=NO, RESOURCE=LP65
FORM	START-SPOOL=NO
FROM=NP/L	FROM=-NP/L, TO=NP/L, DEVICE=REMOTE, T, USER, DEFER, TAPE, VOLUME, START-SPOOL=n/NO, SECTION
FROM=-NP/L	FROM=NP/L, TO=N, TO=NP/L, DEVICE=REMOTE, T, USER, DEFER, START-SPOOL=n/NO, TAPE, VOLUME, SECTION
HEADER=S	SPACE=E/A/I/NO, HEADER=(T,D,P), CONTROL=LINEMODE/LOG/PHYS/ TRANSPARENT, START-SPOOL=NO, CCPOS, PAGECC=NO
HEADER=(T,D,P)	SPACE=E/A/I/NO, HEADER=S, CONTROL=LINEMODE/LOG/PHYS/ TRANSPARENT, START-SPOOL=NO, CCPOS, PAGECC=NO
HDRNUM	START-SPOOL=NO
IMAGE	DEVICE=REMOTE, T, USER, DEFER, PRINTER=136, START-SPOOL=NO, RESOURCE=LP65

Cataloged file name	EAM file number, *, *SYSOUT, *SYSLST, *LIB START-SPOOL=CLOSE/n/NO
LINES	SPACE=E/A/I/NO, CONTROL=LINEMODE/PHYS/TRANSPARENT, CCPOS START-SPOOL=NO
LOCK=YES	DEVIN=TAPE, START-SPOOL=n/NO, *LIB
LOOP	DEVICE=REMOTE, T, USER, DEFER, START-SPOOL=NO
LOOP=(,loop2)	DEVICE=CENTRAL(ND)/REMOTE/RSO-STATION, T, USER, DEFER CONTROL=LOG/LINEMODE/TRANSPARENT, PRINTER=136, START-SPOOL=NO, RESOURCE=LP65
PAGECC=NO	BINARY=YES, HEADER=S/(T,D,P), DEVICE=REMOTE/RSO-STATION, USER, T, DEFER, CONTROL=LOG/TRANSPARENT/LINEMODE, CCPOS, PRINTER=136, START-SPOOL=NO, RESOURCE=LP65
PNAME	START-SPOOL=NO
PRINTER=136	DEVICE=REMOTE/RSO-STATION, T, USER, DEFER, CHARS, DIA, IMAGE, CONTROL=LOG/PHYS/LINEMODE, COPIES=(,N2), START-SPOOL=NO, FOB, LOOP=(,loop2), CHARS-POOL, ROTATION=n/YES, RESOURCE=LP65
PRIORITY	START-SPOOL=NO
RESOURCE=LP65	DEVICE=CENTRAL(ND)/CENTRAL(HP)/RSO-STATION/REMOTE, T, USER, DEFER, CHARS, IMAGE, CONTROL=LOG/PHYS/TRANSPARENT/LINEMODE, PRINTER=136, START-SPOOL=NO, LOOP=(,loop2), CHARS-POOL, FOB, PAGECC=NO, ROTATION=n/YES, SPACE=NO
RETPD	DEVIN=TAPE, DEVICE=CENTRAL(ND)/CENTRAL(HP), START-SPOOL=n/NO, DESTINATION, DEVICE=RSO-STATION, SECTION
ROTATION=n/YES	DEVICE=CENTRAL(ND)/REMOTE/RSO-STATION, T, USER, DEFER, CONTROL=LOG/LINEMODE/TRANSPARENT, PRINTER=136, START-SPOOL=NO, RESOURCE=LP65
SECTION	FROM=NP/L, FROM=-NP/L, TO=N/NP/L, DEVICE=REMOTE, USER, TAPE, T, DEFER, VOLUME, RETPD, START-SPOOL=NO, FAMILY=YES
SHIFT	START-SPOOL=NO
SPACE=1/2/3	SPACE=E/A/I/NO, CONTROL=TRANSPARENT/PHYS, START-SPOOL=NO, CCPOS

SPACE=E/A/I	SPACE=1/2/3/NO, BINARY=YES, HEADER=S/(T,D,P), LINES, CONTROL=LINEMODE, START-SPOOL=NO
SPACE=NO	SPACE=1/2/3/E/A/I, BINARY=YES, HEADER=S/(T,D,P), LINES, T, CONTROL=LINEMODE/LOG/NO, DEVICE=REMOTE/CENTRAL/CENTRAL(ND)/CENTRAL(HP), USER, DEFER, RESOURCE=LP65
STARTNO	START-SPOOL=NO
START-SPOOL=n	Cataloged file name, EAM file number, *, *SYSOUT, DESTROY, DELETE-FILE=DESTROY, LOCK=YES, DEVIN=TAPE, FROM=NP/L, *LIB FROM=-NP/L, TO=N, TO=NP/L, TAPE, VOLUME, RETPD, START-SPOOL=CLOSE/IMMEDIATE/NO
START-SPOOL=CLOSE	Cataloged file name, EAM file number, *, *LIB, START-SPOOL=IMMEDIATE/NO/n
START-SPOOL=IMMEDIATE	START-SPOOL=n/CLOSE/NO,
START-SPOOL=NO	Only compatible with *SYSOUT and *SYSLST
T	FROM=NP/L, FROM=-NP/L, TO=N/NP/L, DEVICE=CENTRAL/CENTRAL(ND), DEVICE=CENTRAL(HP)/RSO-STATION, USER, CHARS, DIA, IMAGE, CONTROL=LOG/PHYS/LINEMODE/TRANSPARENT, COPIES=(,N2), LOOP, LOOP=(,loop2), PRINTER=136, START-SPOOL=NO, FOB, CHARS-POOL, ROTATION=n/YES, CCPOS, PAGECC=NO, *LIB, DESTINATION, SECTION, RESOURCE=LP65, SPACE=NO
TAPE=YES/T9P/T9G/T1600/T6250/TAPE-C1/...	DEVIN=TAPE, DEVICE=CENTRAL(ND)/CENTRAL(HP)/RSO-STATION, START-SPOOL=n/NO, DESTINATION, SECTION
TEXT	START-SPOOL=NO
TO=N	FROM=-NP/L, TO=NP/L, DEVICE=REMOTE, T, USER, DEFER, START-SPOOL=n/NO,
TO=NP/L	FROM=NP/L, FROM=-NP/L, TO=N, DEVICE=REMOTE, T, USER, DEFER, START-SPOOL=n/NO
TRANSLATION-TABLE	DEVICE=CENTRAL/CENTRAL(ND)/CENTRAL(HP)/REMOTE, USER, DEFER, RESOURCE=LP65, ROTATION, FOB, CHARS-POOL, PAGECC=NO, T
TRAY	DEVICE=CENTRAL/CENTRAL(ND)/CENTRAL(HP)/REMOTE, USER, DEFER, DESTINATION=pool, T, RESOURCE, ROTATION=LP65, FOB, PAGECC=NO, CHARS-POOL

TRLNUM	START-SPOOL=NO
USER	FROM=NP/L, FROM=-NP/L, TO=N/NP/L, DEVICE=CENTRAL/CENTRAL(ND), DEVICE=CENTRAL(HP)/RSO-STATION, T, CHARS, DIA, IMAGE, LOOP, CONTROL=LOG/PHYS/LINEMODE/TRANSPARENT, COPIES=(,N2), *LIB, LOOP=(,loop2), PRINTER=136, START-SPOOL=NO, FOB, CHARS-POOL, ROTATION=n/YES, SECTION, RESOURCE=LP65, CCPOS, PAGECC=NO, DESTINATION, SPACE=NO
VOLUME	DEVIN=TAPE, DEVICE=CENTRAL(ND)/CENTRAL(HP), START-SPOOL=n/NO, DESTINATION, DEVICE=RSO-STATION, SECTION
*	Cataloged file name, EAM file number, DESTROY, DELETE-FILE= DESTROY, DEVIN=TAPE, START-SPOOL=n/CLOSE/NO, *LIB
*LIB	Cataloged file name, EAM file number, *, *SYSOUT, *SYSLST, LOCK=YES, DEVICE=REMOTE, T, USER, DEFER, START-SPOOL=n/ CLOSE/NO
*SYSLST	Cataloged file name, EAM file number, *LIB, DESTROY, DELETE-FILE=DESTROY
*SYSOUT	Cataloged file name, EAM file number, *LIB, DESTROY, DELETE-FILE=DESTROY

## Examples

### Example 1:

The following two commands are issued in an interactive job:

```
(IN) PRINT FS1 _____ (01)
(OUT) % SCP0810 SPOOLOUT OF FILE ':Z:$QM123456.FS1' ACCEPTED: TSN: 3286,
      PNAME: 'NEUMAYER'
(IN) STA L
(OUT) NAME          TSN TYPE          PRI      CPU-USED CPU-MAX ACCOUNT#
      3127 3 DIALOG  240          6.6504  32767 A4273V7T
      NAME          TSN TYPE          PRI  SIZE COPIES CPU-USED RTSN
      NEUMAYER 3286 5 PR  255    15          0  0.9401  3127

(IN) PRINT PRIM.TEST, DEVICE=REMOTE, T=STATION1 _____ (02)
```

- (01) The PRINT command is used to generate a spoolout job which is executed independently of the initiating job (TSN=3127) and is given TSN 3286. The user then obtains information on the status of the job with the aid of the STATUS command. TSN 3286 is currently spooling out (TYP=5).
- (02) The file PRIM.TEST is to be output on the batch terminal with the terminal identifier STATION1, on which the following message appears:

```
1981.1717 R003 JOB COMPLETE SOUT1999 ARM      ARM
```

The job name is therefore "SOUTtsn". The file PRIM.TEST is then printed out on the batch terminal.

*Example 2:*

The following commands are given in a batch job:

```
.  
/PRINT DAT, ERASE, SPACE=E, COPIES=3 _____ (01)  
/PRINT TEST.DAT., HEADER=S _____ (02)  
/PRINT (FILE1, FILE2, FILE3), FORM=STDWA4 _____ (03)  
/PRINT (A, A), ERASE _____ (04)  
.
.
```

- (01) File DAT is to be printed out four times and then deleted. The file contains Siemens feed control characters.
- (02) All files whose names start with "TEST.DAT." are to be printed, in each case with a standard header line.
- (03) The files FILE1, FILE2 and FILE3 are to be printed on white paper (DIN A4).
- (04) File A is printed once only, and then deleted. For duplicate printing, the operand COPIES=1 must be specified.



*Example 3:*

The file FILE is dived up into three print pages by SPOOL, with a maximum of 64 lines per page:

Page 1	Page 2	Page 3
<pre> Start      1            2            .            .           20            .            . Line      64         </pre>	<pre> Line      65            .            .            .            .            . Line     128         </pre>	<pre> Line     129            .            . End      150         </pre>

a) /PRINT FILE, FROM=20L      Print FILE from line 20

Page 1	Page 2	Page 3
<pre> Line      20            .            .            .            .            . Line      83         </pre>	<pre> Line      84            .            .            .            .            . Line     147         </pre>	<pre> Line     148            .           149 End      150         </pre>

A page feed is initiated after line 83 (=64+19) and 147 (=128+19).

b) /PRINT FILE, FROM=1, TO=3      Print FILE from page 1 to page 3

The entire file is printed out:

c) /PRINT FILE, FROM=-1      Print the last page of FILE

Page 3
<pre> Line     129            .            . End      150         </pre>

Output consists not of the last 64 lines of FILE, but just the last page, page 3.

## PRIORITY      Modify job or run priority

Application group: Job control (page 22 ff.)

### Command description

The PRIORITY command modifies the scheduling or run priority of a job and/or enables it to be started immediately.

The job scheduling priority determines the urgency (relative to other jobs) for starting a batch job. It has no effect on the further processing of the job. The highest assignable priority is defined in the job class definition and in the JOIN entry (see also SHOW-JOB-CLASS and SHOW-USER-ATTRIBUTES commands), as is the authorization for the EXPRESS operand (or IMMEDIATELY option in the case of START).

The job is addressed via its TSN or monitoring job variable. The latter case is only possible for users with the JV software product.

### Format and operand description

Operation	Operands
<pre>{ PRIORITY } { PRI }</pre>	<pre>{ tsn } , { p } { MONJV=jvname } , ( [p] , EXPRESS )</pre>

**tsn**                      Task sequence number of the job.

**EXPRESS**                Specifies that the batch job is to be started immediately.  
 This entry has no effect on the further processing of the job.  
 The authorization for entering EXPRESS is defined in the JOIN entry.

**MONJV=jvname**            Name of the job variable specified in the LOGON or ENTER command.  
  
 The command is rejected if the job variable was not defined in the LOGON or ENTER command.



MONJV is only available with the JV software product (see also "Job Variables" manual [11]).

p Job or run priority.

Job priority:  $\text{MAXIMUM} \leq p \leq 9$

Run priority:  $\text{MAXIMUM} \leq p \leq 255$ .

The MAXIMUM values are defined in the job class definition and the JOIN entry, and can be displayed on the screen with the commands SHOW-JOB-CLASS or SHOW-USER-ATTRIBUTES. If the values for "rprio" in the job class definition and the JOIN entry do not match, the optimum value for the user is selected.

*Example:*

```
/PRIORITY 6785,255
```

The job with TSN 6785 is given priority 255. This alteration can be checked by means of the STATUS command.

## PROCEDURE      Define procedure file attributes

Application group: job control (page 22 ff.)

### Command description

The PROCEDURE command identifies a file or a library member as a procedure file.

The PROCEDURE command lists the symbolic names used in the procedure file in the form of keywords and/or positional operands which always begin with the prefix "&".

The number of positional operands and/or keyword operands is limited. The following applies:

number of positional operands + 2 \* number of keyword operands ≤ 255.

It is possible to declare a prefix for the symbolic names in input data.

It is also possible to control the logging of procedure execution by means of an operand. Together with this operand, you can specify that execution of the procedure cannot be interrupted by hitting the K2 key in order to enter commands in interactive mode. When the K2 key is pressed, the system queries whether or not the job is to be terminated. This feature is specified separately for each procedure level in the case of nested procedures. This does not affect the starting of an STXIT routine assigned to event class ESCPBRK within a program.

The PROCEDURE command is the first record (procedure header) in a procedure file. Subsequent records contain either commands or input data. A distinction must be made between the symbolic names which occur in the commands and those occurring in input data. Symbolic names in commands also begin with the prefix "&", whereas symbolic names in the input data are distinguished by a prefix declared with the SUBDTA operand. During execution of the procedure, the symbolic names are replaced by current values.

The last record in a procedure file is the ENDP command.

A procedure (procedure file) is started with the DO or CALL command.

An invalid command call in a procedure file usually triggers a SPIN-OFF (exception: see list at end of description). This means that a branch is made to the next ABEND, ABORT, STEP or LOGOFF command, all other commands being ignored. Whereas the ABEND or LOGOFF terminates the job and the ABORT command terminates the procedure, the procedure is continued after the STEP command.

DO procedure:      If the invalid command call is followed by none of the above commands, procedure level 0 (= system level) is assumed.

CALL procedure:      If the invalid command call is followed by none of the above commands, a branch is made to the last ABEND, ABORT, LOGOFF

or STEP at a lower procedure level; otherwise procedure level 0 is assumed.

The PROCEDURE command must occur once only in a procedure file. It may not occur in the procedure file at the same time as the LOGON or RESTART command.

The following applies to records which come after a PROCEDURE command:

- a) A symbolic name begins with the "&" or SUBDTA character and ends with a period. This period is also replaced when the current values are substituted.
- b) The period may be omitted if the symbolic name does not contain any further characters.

c) *Example:*

```

Call          : /DO name, (ID=CC, COMP=FOR1)

Proc. file   : /PROC, (&ID=, &COMP=)
               :
               : /EXEC $&IDENTIF..&COMP
               :
               :           ↓       ↓       ↓
Substitution: /EXEC $   RZ       . FOR1
               (/EXEC $RZ.FOR1)

```

**SPIN-OFF**

An invalid command name given in a procedure always activates a SPIN-OFF. This is an error-triggered branch to the next STEP, ABEND, ABORT or LOGOFF command, the intermediate commands being ignored. An invalid command operand or an invalid assignment for the operand usually triggers a SPIN-OFF. The table below shows whether or not a SPIN-OFF is triggered.

Operand error in command	Next command is executed	Branch to system level
APPLICATION	X	
AUDIT	X	
BCNTRL	X	
CANCEL <sup>1)</sup>	X	
CONNECTION	X	
HELP	X	
LOGON	X	
MRSSTA ON <sup>2)</sup>	X	X
OPTION	X	
PRINT	X	
PRIORITY	X	
PROCEDURE <sup>3)</sup>		X
PUNCH	X	
SET-SS-OPTIONS	X	
SHOW-DISK-STATUS	X	
SHOW-JOB-CLASS	X	
SHOW-SPOOL-CHARACTER-SET	X	
SHOW-SPOOL-DEVICES	X	
SHOW-SPOOL-FORMS	X	
SHOW-TAPE-STATUS	X	
SPARAM	X	
STAM	X	
SYSTATUS	X	
TCHNG	X	

- 1) Applies only if a TSN that cannot be found under the user ID is specified in a command call.
- 2) Applies only if the ENDON command which forms part of the ON construct cannot be found.
- 3) Applies only if the values transferred from the **DO** or **CALL** command do not match the structure of the operands in the **PROCEDURE** command. In this case the procedure is terminated and procedure level 0 is entered when the procedure is a **DO** procedure. For **CALL** procedures, a branch is made to the last **STEP** command.

**Format and operand description**

Operation	Operand
$\left. \begin{array}{l} \text{PROCEDURE} \\ \text{PROC} \end{array} \right\}$	$\left[ \begin{array}{l} \text{N[X]} \\ \text{C[X]} \\ \text{D[X]} \\ \text{A[X]} \end{array} \right] [, (\text{symbolparam} \left\{ \begin{array}{l} =[\text{default}] \\ =[\text{default}] \end{array} \right\} [, \dots]) [, \text{SUBDTA}=\left\{ \begin{array}{l} \& \\ @ \\ \# \\ * \\ \$ \end{array} \right\} ]]$

**N** Procedure file processing will not be logged to system file SYSOUT.

**C** Procedure file commands are to be logged to SYSOUT when executed.

**D** Data processed within the procedure file is to be logged to SYSOUT.

**A** Commands and data processed within the procedure file are to be logged to SYSOUT.

**X** Procedure execution cannot be interrupted by hitting the K2 key in order to enter commands in interactive mode.

**symbolparam** Symbolic name from the procedure file.  
It starts with the prefix "&", followed by a letter and up to 253 further alphanumeric characters. Names beginning with "&&" are not viewed as symbolic names, i.e. they are not replaced by current values.

**=default** Current value for the symbolic name.  
This assignment is used as a default value if the symbolic name is not cited in the **DO** or **CALL** command. Length of "default" = 254 characters.  
"default" must not be a complete command (with slash).

**SUBDTA={...}** Specifies a prefix for identifying symbolic names in records with input data.  
  
In the **PROCEDURE** command, however, these symbolic names begin with the prefix "&" (see "symbolparam").  
  
To avoid undesired substitutions, care must be taken to select the appropriate prefix.  
  
Names with double prefixes are not viewed as symbolic names. When the procedure is executed, the compound character is reduced to a single character and assigned to the subsequent characters.

**Examples:***Example 1:*

A procedure file contains the following commands:

```
/PROCEDURE A, (&A, &B), SUBDTA=@
.....
/SYSFILE SYSDTA=(SYSCMD)
/EXEC $LMR
MODLIB=X.OML.@A
ADD OBJMOD=@B, SOURCE=*
END
/FSTAT X.OML.&A,ALL "OUTPUT CATALOG ENTRY"
/ENDP
```

**The call**

```
/DO PROCEDURE FILE, (ALPHA, BETA)
```

is processed as follows:

```
/EXEC $LMR
MODLIB=X.OML.ALPHA
ADD OBJMOD=BETA, SOURCE=*
END
/FSTAT X.OML.ALPHA,ALL "OUTPUT CATALOG ENTRY"
```

*Example 2:*

The following procedure file is generated for interactive mode:

```
/PROC N, (&EIN, &AUS=), SUBDTA=*
/SYSFILE SYSDTA=(SYSCMD)
/SETSW ON=(4,5)
/EXEC $EDT
@READ '*IN'
@PRINT & N
.....
@WRITE '*OUT'
ALT
/ENDP
```

The input and output files for the file editor EDT are not defined until in the DO command or during procedure execution.

For further examples, see the SKIP and BREAK commands.



## PSWORD      Protect user ID by means of password

Application group: Job control (page 22 ff.)

### Command description

The PSWORD command allows you to create a LOGON password or delete or change an existing password. The LOGON password may be entered as a character constant C'....' or as a hexadecimal constant X'.....'; with a maximum length of 8 bytes unless otherwise specified by the system administrator.

- A LOGON password protects the user ID against unauthorized access.
- Defining, changing or deleting a LOGON password requires authorization issued by the system administrator in the JOIN entry. You can request information concerning this authorization by using the SHOW-USER-ATTRIBUTES command (PSWORD output field). The PSWORD field entries denote the following:

PSWORD = YES: You are authorized to define a LOGON password and/or to delete an existing LOGON password.

PSWORD = MOD: You are only authorized to modify an existing password.

PSWORD = NO: You are not authorized to define, delete or modify a password.

In the job log passwords are overwritten with the character "P".

- For reasons of data protection, the maximum length should always be used when defining or modifying a LOGON password.

**Format and operand description**

Operation	Operands
PSWORD	$\left\{ \begin{array}{l} [\text{oldpswrd}], \text{newpswrd} \\ \text{oldpswrd} \end{array} \right\} [, \left\{ \begin{array}{l} \text{PUBLIC-VOLUME-SET} \\ \text{PVSID} \end{array} \right\} = \left\{ \begin{array}{l} \text{*HOME} \\ \text{catid} \end{array} \right\} ]$

**oldpswrd** Denotes the existing password.  
 When "oldpswrd" is entered without the subsequent entry "newpswrd", the existing (old) password is deleted (overwritten with X'0000000000000000').

**newpswrd** Denotes the new password. The corresponding entry in the TSOSJOIN file is overwritten. The entry X'0000000000000000' is interpreted as password deletion, unless the operating system uses password encryption. In this case, X'00...0' is encrypted and forms the password. When ",newpswrd" is entered (without the entry "oldpswrd"), a LOGON password is specified if no such password was previously available.

**PUBLIC-VOLUME-SET**

Identifies the pubset on which the TSOSJOIN file with your JOIN entry is located.

**PVSID**

=\*HOME Home pubset; default value.

=catid Pubset catalog ID.

**Examples:**

**Create LOGON password:**

```
(IN) LOGON XYZ,ABR07MAN
      . . .
(IN) PSWORD ,C'LORELEY1' _____ (01)
(IN) LOGOFF BUT
      . . .
(IN) LOGON XYZ,ABR07MAN
(OUT) % JMS0151 PLEASE ENTER PASSWORD.
(IN) C'LORELEY1' _____ (02)
(OUT) % JMS0066 JOB ACCEPTED ON 87-09-07 AT 16:10, TSN = 4277
```

**Change LOGON password:**

```
(IN) PSWORD C'LORELEY1',C'LORELEY2' _____ (03)
(IN) LOGOFF BUT
      . . .
(IN) LOGON XYZ,ABR07MAN,C'LORELEY2'
(OUT) % JMS0066 JOB ACCEPTED ON 87-09-07 AT 16:18, TSN = 4281
```

**Delete LOGON password:**

```
(IN) PSWORD C'LORELEY2' _____ (04)
(IN) LOGOFF BUT
      . . .
(IN) LOGON XYZ,ABR07MAN
(OUT) % JMS0066 JOB ACCEPTED ON 87-09-07 AT 16:25, TSN = 4293
```

- (01) The PSWORD command defines the password "LORELEY1" for the JOIN entry of user ID "XYZ".
- (02) The user is requested to enter the password. The input is blanked; the job is accepted.
- (03) The existing password "LORELEY1" is changed to "LORELEY2"; the entry in the LOGON command is changed accordingly.
- (04) The password "LORELEY2" is deleted; there is no longer a password for user ID "XYZ".

## PUNCH Output file

Application groups:

Job control (page 22 ff.)

File processing (page 26 ff.)

### Command description

The PUNCH command allows you to output files to floppy disk. By specifying the appropriate operands you can use the following optional functions at the same time:

Operand	Function
pathname prefix[name] eamno * *SYSOPT	} Denote files to be output.
START-SPOOL	Determines the printing time for SYSOPT.
ERASE	Deletes the file to be output after output (positional operand).
DESTROY	Overwrites the file with binary zeros (positional operand).
DELETE-FILE	Deletes or overwrites the file with binary zeros after printing.
LOCK	Locks the file to be output against updating, until output has finished or the job is ended.
RETPD	Declares a retention period for the file.
PRIORITY	Specifies a run priority for the spoolout job.
PNAME	Specifies a job name for the spoolout job.
DEVICE	Specifies the device type.
STARTNO ENDNO	Limits output to a specific subarea of the record to be printed.
VOLUME	Specifies output to SCRATCH tapes.
LOG	Generates a logging file (for output to floppy disk).

The PUNCH command specifications and device selection data (defaults) are entered in the Spool Control Block (SCB) which is added to one of the spoolout queues (local spoolout) and managed as a separate job (with its own TSN).

You can use the STATUS command to trace the processing of your spoolout job, and you can use the PRIORITY and CANCEL commands to change the run priority of your job or terminate it.

No more than 99 volumes can be assigned to one file. There is no restriction for file groups.

A new volume is requested for every PUNCH command. However, the files specified in the PUNCH command are written to one volume and, if necessary, its continuation volumes.

The LOGON command generated by specifying SKEL=US contains no TIME operand. Before the floppy disk contents are read in, you must specify in the LOGON command a suitable CPU time for the current job. The spoolin job is aborted if the CPU time available for the current job is less than the CPU time assigned by default to the spoolin job (see also "SPOOL System" manual [19]).

If the VOLUME operand is omitted, the OWNERID=N operand must be specified. Output is then to so-called SCRATCH floppy disks.

If, for example, you have specified 2 volume serial numbers and the output takes up more than 2 floppy disks, SCRATCH floppy disks are required. The volume serial numbers of the SCRATCH floppy disks are written to the logging file if LOG=YES was specified in the PUNCH command.

The operator receives a message containing a list of all volume serial numbers and prompting him to mount the appropriate volume(s). The following situations may arise:

- a) If the correct volume is mounted, the write procedure can be started normally and it is executed.
- b) If the correct volume cannot be found, the operator responds with "NO" and sends you a message. The spoolout job is aborted.
- c) If the operator responds affirmatively but has mounted the wrong volume, he receives another message requesting him to mount the correct volume. He can either repeat or terminate this procedure.

If LOG=YES is specified, these messages are recorded in the logging file.

If the spoolout job has to be interrupted, the operator can have the run repeated immediately or later. Processing can be started from the beginning or from the last floppy disk processed. If a number of files are specified, output is to a SCRATCH floppy disk in the event of a restart - even if a list of volume serial numbers was specified. The sector length is always 128 bytes if DEVICE=DISKETTE or DEVICE=FD3170 is specified. A sector length of 128/256/512/1024/2048 bytes is possible if DEVICE=FD3171 is specified. In both cases records which are too long are truncated. Shorter records are padded with blanks (X'40'). The sector length is noted in the HDR1 label. A file record is always entered in a (whole) sector on the floppy disk. Before output the volume is checked to ensure that writing to it is possible. The following conditions must be satisfied:

- a) The access restriction indicators must not be set in either the VOL1 label or the HDR1 labels.
- b) There must be no write protection.
- c) The retention period must have expired.

A file can be protected by means of the RETPD, ACCESS, WRITEPR and BYPASS operands.

At output time the following user data is entered on the volumes:

- a) The user ID (from the LOGON command) is entered in the volume label (VOL1) and access restriction is set in accordance with the specifications for the individual files (even if access restriction is for one file only, the whole volume is locked).
- b) For each file, the following data from the PUNCH command is entered in the HDR1 label:
  - user name (NAME operand)
  - address of the last sector of a file (SECTORS operand)
  - date of release (obtainable from the RETPD operand)
  - skip (BYPASS operand)
  - access restriction (ACCESS operand)
  - write protection (WRITEPR operand)

## Formats and operand descriptions

### Format 1: Outputting one file to floppy disk

Operation	Operands
PUNCH	<pre> {   pathname   prefix[name]   eamno   *   *SYSOPT[ , START-SPOOL={ IMMEDIATE                         CLOSE                         NO } ] }  [ , {   ERASE   DESTROY } ]  [ , DELETE-FILE={ NO                   DESTROY                   ERASE } ]  [ , PNAME=pname ]  [ , LOCK={ NO            YES } ]  [ , RETPD=days ]  [ , PRIORITY=p ]  [ , STARTNO=byteno ] [ , ENDNO=byteno ]  , DEVICE={ DISKETTE            FD3170            FD3171 }  [ , FDTYPE={ FD1S1D              FD2S1D              FD2S2D } ]  [ , VOLUME={ vsn              { (vsn1 , . . . ) } } ] </pre>

Operation	Operands													
PUNCH (continued)	[,RECSIZE=n]  [,SECTORS=number]  [,NAME=name]  [,BYPASS= <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>YES</td></tr> <tr><td>NO</td></tr> </table> ]  [,ACCESS= <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>YES</td></tr> <tr><td>NO</td></tr> </table> ]  [,WRITEPR= <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>YES</td></tr> <tr><td>NO</td></tr> </table> ]  [,SKEL= <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>US</td></tr> <tr><td>NS</td></tr> <tr><td>NO</td></tr> </table> ]  [,OWNERID= <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>YES</td></tr> <tr><td>NO</td></tr> </table> ]  [,LOG= <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>YES</td></tr> <tr><td>NO</td></tr> </table> ]	YES	NO	YES	NO	YES	NO	US	NS	NO	YES	NO	YES	NO
YES														
NO														
YES														
NO														
YES														
NO														
US														
NS														
NO														
YES														
NO														
YES														
NO														



## Operand description (keyword operands in alphabetical order)

pathname stands for:  $[:catid:][\$userid.] \left\{ \begin{array}{l} \text{file} \\ \text{group} \left\{ \begin{array}{l} (*abs) \\ (+rel) \\ (-rel) \end{array} \right\} \end{array} \right\}$

A number of pathnames (up to 11) can be specified in parentheses and separated by commas.

catid	Catalog ID of the pubset on which the file is stored. Default value: the catalog ID assigned to the user ID (JOIN entry).
userid	User ID to which the catalog is assigned. Default value: user ID from the LOGON command.
group	Name of the file generation group from which the file (generation) is to be printed. The name must not exceed 34 characters in length and must not be partially qualified (See "DMS" manuals [8], [9]).
abs	Absolute generation number of the file generation to be printed. "abs" is an element of the set (1, 2, ..., 9999).
rel	Relative generation number of the file generation to be printed. "rel" is an element of the set (1, 2, ..., 99) and refers to a base value (see the BASE field in the output of the FSTAT command). The following applies: rel = abs - base value.
file	Fully or partially qualified file name, fully-qualified name of a file generation, or name of a temporary file.  The PUNCH command is rejected if <ul style="list-style-type: none"> <li>– "file" is a newly cataloged file into which nothing has yet been written;</li> <li>– "file" has already been opened in output mode.</li> </ul>

*Notes for temporary (user) files:*

A temporary file begins with a character defined at system generation time for prefixing the file names of temporary files. Temporary files are stored according to job and deleted when the job is finished. A PUNCH command to a temporary file is always executed automatically with LOCK=YES,ERASE. This prevents a temporary file from being prematurely deleted by LOGOFF instead of when printing is finished.

The temporary file is also deleted if the spoolout job has abnormally terminated (e.g. by means of the CANCEL command). For information on temporary files, see also the manual "DMS Introductory Guide and Command Interface" [8].

prefix[name]	The specified temporary (user) file is output. All temporary (user) files are output if only "prefix" is specified.  "prefix" is a character defined at system generation time as a prefix for the file names of temporary files. "name" is any (file) name with a length of up to 30 characters.
eamno	Number of an EAM file (see the manual "DMS Introductory Guide and Command Interface" [8]). When more than one number is specified, they must be separated by commas or enclosed in parentheses.
*	Indicates the EAM object module file of the job.
ACCESS=NO	Specifies that the file is to be locked for access. An appropriate flag is set in the volume label (VOL1). Once a file is locked, the whole volume is locked to access for both offline mode and teleprocessing. Access is only possible using the RFD command (see the "SPOOL System" manual [19]) and the PNCH macro under the user ID entered in the volume label or under the ID of the system administrator. The user ID is always entered in the volume label (VOL1) by a PNCH macro.
=YES	Default value: The file is not locked.
BYPASS=YES	Specifies that the file is to be bypassed when it is reread (RFD command, see the "SPOOL System" manual [19]. Accordingly a flag (C'B') is set in the header label (HDR1) of "filename".
=NO	The file will not be bypassed when the floppy disk is reread.

DELETE-FILE	
=ERASE	Same function as ERASE.
=DESTROY	Same function as DESTROY.
= <u>NO</u>	The file is not deleted after output (exception: SYSOPT and EAM object module file *); default value.
DESTROY	Indicates that after the file is printed, its catalog entry and data are overwritten with X'00...0' (does not apply to EAM files and SYSOPT). Default value: the file is not overwritten after print  If the file to be printed has the catalog entry ACCESS=READ, the ERASE, DESTROY and DELETE-FILE operands are ignored.  ERASE and DESTROY are positional operands; note the position in the command format.  ERASE or DESTROY must not be specified in conjunction with the *SYSOPT operands.
DEVICE	Designates the device type.
=DISKETTE	Output is to a scratch floppy disk.
=FD3170	Output is to an FD3170 Floppy Disk I/O Unit.
=FD3171	Output is to an FD3171 Floppy Disk I/O Unit.
ENDNO	Specifies a byte number (record column), up to which the records are output. Output ends with the specified byte. (The bytes of a record are numbered from left to right, beginning with 0.)
=byteno	Number of the data byte; $0 \leq \text{byteno} \leq 32767$ . Default value: – end of record – byteno = 2048 for an EAM File or SYSOPT.

**ERASE** Deletes the file as soon as output is ended. You must have write access to the file. Default value: the file is not deleted after printing.

If the file to be printed has the catalog entry ACCESS=READ, the ERASE, DESTROY and DELETE-FILE operands are ignored.

ERASE and DESTROY are positional operands; note the position in the command format.

ERASE or DESTROY must not be specified in conjunction with the \*SYSOPT operands.

ERASE is ignored if the file to be output belongs to a file generation group.

**FDDTYPE** Designates the floppy disk type for the FD3171 device.

=FD1S1D Single-sided, single density.


=FD2S1D Double-sided, single density.

=FD2S2D Double-sided, double density.

Default values:

FD1S1D, for record lengths ≤ 1024

FD2S2D, for record lengths > 1024

 If floppy disk device type FD3170 was specified, the FDDTYPE operand is not interpreted.

If a scratch floppy disk (DEVICE=DISKETTE) is requested, spoolout attempts to find a floppy disk with sufficient sector length (i.e. a sector length greater than the specified record length).

**LOG**

=YES A logging file is generated by each spoolout job to floppy disk. This file contains all relevant console messages (mount or error messages) and the file names concerned. This takes place regardless of whether or not errors have occurred. The logging file is output to printer as soon as the spoolout job has terminated.

=NO No logging file is generated.

LOCK	
=YES	Locks the file against updating for as long as the spoolout job is in the wait state (TYPE 4) or if it is terminated. During this time period the file can only be read. (For TYPE 4 see the STATUS LIST command). File protection resulting from LOCK=YES is also retained if output does not begin until the next session. The file is automatically protected from the beginning to the end of job or until the job is cancelled. LOCK=YES must not be used together with * or *SYSOPT.
=NO	The file is not protected while the spoolout job is in the wait state. The file can be deleted or updated until the time output begins.  The default value for the LOCK operand is defined in the SPOOL parameter file. Specification of the LOCK operand is not permitted if the file is to be output to tape.
NAME=name	Specifies the name to be written to the HDR1 label of the file. "name" may be up to 8 characters long. The system does not check for duplicate user names. Default value: the first 8 characters of the cataloged file name.
OWNERID= <u>YES</u>	The owner identifier field in the VOL1 label contains the user ID.
=NO	The field contains blanks only. If the VOLUME operand is omitted, the OWNERID=N operand must be specified, in which case spoolout will be to SCRATCH floppy disks.
PNAME=pname	Job name for the spoolout job. "pname" can consist of up to 8 characters of the set (A,...,Z,0,...,9,@,#,\$,.,-), but it must not begin or end with one of the last two characters. The job name appears in the output of the STATUS command. If this operand is omitted, the job name from the LOGON command is assumed instead.
PRIORITY	Determines the urgency of the spoolout job relative to other spoolout jobs.

=p Run priority.  $\text{MAXIMUM} \leq p \leq 255$ . The value for MAXIMUM is defined in the JOIN entry and can be interrogated with the SHOW-USER-ATTRIBUTES command.  
If invalid values are specified for "p" or if the PRIORITY operand is omitted, the operating system assumes the priority of the generating job.

RECSIZE Specifies the maximum record length.

=n "n" is copied to the HDR1 label of the of the floppy disk. The following values apply for "n":  
DEVICE=FD3170:  $1 \leq n \leq 128$ .  
DEVICE=FD3171 and FDTYPE=FD1S1D/FD2S1D :  $1 \leq n \leq 1024$ .  
DEVICE=FD3171 and FDTYPE=FD2S2D :  $1 \leq n \leq 2048$ .

RETPD Defines a retention period for the file in days, i.e. a period during which the file may be read, but not updated or erased.

=days Retention period in days (0..999). Default value: days=0, i.e. the file may be updated immediately.

SECTORS=number Number of sectors (= records) to be reserved at the end of the file. If there is not enough space available for these sectors on the volume, they are only reserved up to the end of the volume. Irrespective of the required number, all sectors are reserved through the end of the particular track.  
Default value: "number" = 0.

SKEL=US The file is prepared as an ENTER job. The following records (commands) are inserted at the front or the back of the file:

```
/LOGON userid,accountno,password  
/DATA filename,SAM or ISAM
```



```
/END  
/LOGOFF
```

- These insertions must be added so that the file can be read by the BS2000 operating system. "userid", "accountno", and "password" entries are obtained from the LOGON for the current job. However the password is only copied if the operating system has not been generated with password encryption feature. In this case the first record  
/LOGON userid,accountno is inserted at the beginning of the file, and you must add the password in offline mode.
- =NS** Same effect as SKEL=US except that the first record to be prefixed is only /LOGON; "userid", "accountno", and "password" (data security) are omitted. That information must be added offline by the user.
- =NO** No records are added to the file; default value.
- STARTNO** Specifies a byte number (record column). The records of the file are output starting at the specified data byte. (The bytes of a record are numbered from left to right, beginning with 0.)
- =byteno** Number of the data byte;  $0 \leq \text{byteno} \leq 32767$  Default value: byteno = 1; output begins with byteno 1, i.e. with the 2nd data byte.
- If "byteno" is greater than the length of the record, the record is not printed.
- ISAM keys and control characters are part of each record. Whether or not they are also printed depends on the STARTNO=... entry.
- START-SPOOL** For system file SYSOPT, determines output time prior to job termination.
- =IMMEDIATE** The spoolout job is initiated immediately; default value.
- =CLOSE** The spoolout job is initiated immediately after the system file is closed.
- Closing a system file:
- a) primary allocation: with the LOGOFF command.
  - b) if assigned to a cataloged file:
    - New SYSDFILE command (change of assignment)
    - LOGOFF command (output is suppressed with LOGOFF NOSPOOL)
    - In a procedure run, when procedure level 0 is reached.

=NO                    A previous definition for START-SPOOL is cancelled.  
Additional operands are rejected.

Switching this START-SPOOL definition is only possible via the intermediate step START-SPOOL=NO. Otherwise, the command will be rejected.

Every spoolout job is given its own TSN.

The operand value IMMEDIATE is not allowed for tape files.

\*SYSOPT                System file SYSOPT is output. If assigned to a cataloged file the latter's contents will be output.

*\*SYSOPT and \**

An EAM file (identified by file number or \*) and the (system) file SYSOPT are deleted at the end of output. The files are also deleted if the spoolout job is terminated abnormally (e.g. with the CANCEL command). Up to 2268 EAM files may be specified in a PUNCH command.

The EAM file \* and \*SYSOPT may be specified together in the same PUNCH command, however they must not be specified together with an EAM file number or a cataloged file.

VOLUME

=vsn                    Specifies the volume serial number of the floppy disk (maximum length: 6 characters).

=(vsn1,...vsn10)

A maximum of 10 VSNs may be specified.  
If the VOLUME operand is omitted, the ONWERID=N operand must be specified, in which case spoolout will be to SCRATCH floppy disks.

WRITEPR=YES           Specifies that the file must not be overwritten. A write protection flag (C'P') is set in the HDR1 label (beginning-of-file label).

=NO                    The file must not be overwritten.



**Format 2:** Output of several files to floppy disk

Operation	Operands
PUNCH	<pre> (pathname, ...)  [ , {   { ERASE     DESTROY   }   DELETE-FILE= { NO                  DESTROY                  ERASE                } } ]  [ , PNAME=pname ]  [ , LOCK= { YES            NO          } ]  [ , PRIORITY=p ]  [ , STARTNO=byteno ] [ , ENDNO=byteno ]  , DEVICE= { (DISKETTE, ...)             (FD3170, ... )             (FD3171, ... )           }  [ , FDTYPE= { (FD1S1D, ...)               (FD2S1D, ...)               (FD2S2D, ...)             } ]  [ , RECSIZE= (n1, n2 ... n10) ]  [ , NAME= ( [name1] , [ , name2] , ... , [names] ) ]  [ , SECTORS= ( [number1] , [number2] , ... , [number] ) ]  [ , RETPD= ( [days] , [ , days2] , ... , [days] ) ]  [ , BYPASS= { { YES                NO              }              ( { YES                  NO                } , ... )            } ] </pre>

Operation	Operands
PUNCH (cont.)	$[ , ACCESS = \left\{ \begin{array}{l} \{ \underline{YES} \} \\ \{ NO \} \end{array} \right\} ]$ $\left( \begin{array}{l} \{ \underline{YES} \} \\ \{ NO \} \end{array} \right) , \dots$ $[ , WRITEPR = \left\{ \begin{array}{l} \{ \underline{YES} \} \\ \{ NO \} \end{array} \right\} ]$ $\left( \begin{array}{l} \{ \underline{YES} \} \\ \{ NO \} \end{array} \right) , \dots$ $[ , SKEL = \left\{ \begin{array}{l} \{ \underline{US} \} \\ \{ \underline{NS} \} \\ \{ \underline{NO} \} \end{array} \right\} ]$ $\left( \begin{array}{l} \{ \underline{US} \} \\ \{ \underline{NS} \} \\ \{ \underline{NO} \} \end{array} \right) , \dots$ $[ , VOLUME = \left\{ \begin{array}{l} \{ vsn \} \\ \{ (vsn, \dots) \} \end{array} \right\} ]$ $[ , OWNERID = \left\{ \begin{array}{l} \{ \underline{YES} \} \\ \{ NO \} \end{array} \right\} ]$ { YES } { <u>NO</u> }

With the exception of the DEVICE, VOLUME and OWNERID operands, it is possible to specify a number of values for one operand. These values are assigned according to their order in the corresponding file in the expression (pathname,...,pathname) (see also example below).

**Operand description** See Format 1: Output of one file to floppy disk.

(pathname,...) Up to 11 file names may be specified (fully or partially qualified file names or fully-qualified names of file generations). The specified files are stored on a volume and any continuation volumes which may be required.

Multiple operand values for the above-mentioned operands are assigned to the corresponding file according to their positions: 1st operand value to 1st file name, 2nd operand value to 2nd file name, etc.

If an operand value is omitted, the comma must nevertheless still be entered and the preceding operand value is accepted. If, however, the first operand value is omitted, the default is used. Exception: If an operand value is omitted in the NAME operand, the first 8 characters of the catalog file name are assumed.

*Example:*

```
PUNCH (DATA, DATB, DATC)
.
.
.
, DEVICE=DISKETTE
, RECSIZE= (90, 91, 96)
, NAME= (MODX, MODY, MODZ)
, SECTORS= (2, 4, 2)
, RETPD= (30, 60, 0)
, BYPASS= (N, Y, N)
, ACCESS= (Y, N, N)
, WRITEPR= (N, N, Y)
, SKEL= (US, N, N)
```

The file whose cataloged name is DATA has the user name MODX;

- its record length is 90 bytes
- 2 sectors are reserved
- the retention period is 30 days
- the file is not bypassed on readin, access is enabled, the file may be overwritten
- the file is made compatible with BS2000.

The file whose cataloged name is DATB has the user name MODY.

- its record length is 91 bytes
- 4 sectors are reserved
- the retention period is 60 days
- the file is not bypassed on readin, access is locked, the file may be overwritten and no compatibility with BS2000 is provided.

Although operand values may be omitted from the sequence, the comma must be written, in which case the preceding value applies:

ACCESS=(Y,N,N) has the same effect as ACCESS=(Y,N,)

When the first value in a sequence is omitted, the default is used:

WRITEPR=(N,N,Y) has the same effect WRITEPR=(,Y)

### *Exceptions*

- For values omitted from the NAME operand, the system always supplies the first 8 characters of the assigned file name, as in Format 2.
- For the RECSIZE operand, the number of values "n" must match the number of file names. Alternatively, only one value "n" may be specified. This value is then assigned to all file names.

**Format 3:** Output of files to pooler tape

This format is only supported for reasons of compatibility; it will not be supported in versions that follow BS2000 V10.

Operation	Operands
PUNCH	<pre> {   pathname   prefix[name]   * } *SYSOPT[ , START-SPOOL={   IMMEDIATE   CLOSE   NO }]  [ , {   ERASE   DESTROY } ]  [ , DELETE-FILE={   NO   DESTROY   ERASE } ]  [ , LOCK={   NO   YES } ]  [ , DEVICE=CENTRAL ]  [ , PRIORITY=p ]  [ , STARTNO=byteno ] [ , ENDNO=byteno ]  , TAPE=POOLER  [ , VOLUME={   SCRATCH   vsn   (vsn, ... ) } ]  [ , NAME={   name   (name, ... ) } ] </pre>

Operation	Operands
PUNCH (continued)	[, RETPD=days]  $\left[ , \text{SKEL} = \left\{ \begin{array}{l} \left[ \begin{array}{l} \text{US} \\ \text{NS} \\ \text{NO} \end{array} \right] \\ \left( \begin{array}{l} \text{US} \\ \text{NS} \\ \text{NO} \end{array} \right), \dots \end{array} \right\} \right]$  [, BLKSIZE=n]

**Operand description (in alphabetical order)**

pathname prefix[name] * SYSOPT START-SPOOL ERASE DESTROY DELETE-FILE LOCK PRIORITY STARTNO ENDNO VOLUME RETPD NAME SKEL	}	For a description of these operands see Format 1: output of a file to floppy disk
--	---	---

**BLKSIZE**            The operand is only allowed for output to pooler tape. Specifies that non-standard blocks are to be processed and defines the buffer length.

=n                     Buffer length in Kbytes; "n" may be any of the following:

- n=0    Default value.  
         Fixed-length records (128 bytes), one record per block.
- n=2    Buffer length 2 Kb
- n=4    Buffer length 4 Kb
- n=8    Buffer length 8 Kb

DEVICE=CENTRAL

Specifies output to tape.

## TAPE=POOLER

Specifies that the file is to be written to a pooler tape in a format which allows for offline transfer to floppy disk. This operand is mandatory.

## VOLUME

=SCRATCH

Specifies that output is to SCRATCH tapes if you have specified the TAPE operand; default setting.

These SCRATCH tapes are provided by the operator.

## =vsn

Volume serial number of the tape to be used for output.

## =(vsn,...)

Up to 4 volume serial numbers may be defined. This operand is ignored if the TAPE operand has been omitted.

The tapes are written in the specified order, but only the number of tapes required. If a specified tape cannot be accessed, SPOOL automatically provides a SCRATCH tape.

If the retention period (RETPD) of a file to be written to a tape specified with VOLUME is longer than that of the file last written to this tape, the file is written to a SCRATCH tape instead of the specified tape.

## RDTFT Request information from TFT and TST

Application group: Interrogation of current values (page 31)

### Command description

The RDTFT command allows you to display status information from the TFT (Task File Table) concerning the files and tape devices currently in use. In addition, you also receive information from the associated TST entry (see the FILE command).

### Format and operand description

Operation	Operand
RDTFT	[FILE=pathname]  [, LINK={link (BLANK, no)}]  [, LINKAGE] [, SECURITY] [, FCB] [, VOLUMES]  [, ALL]

**ALL** Indicates that all the information described above is to be output for each TFT entry specified.



FCB

Requests information on file attributes. The output fields are explained below:

Output field	Values	Description
FCBTYPE	SAM/ISAM/BTAM/ PAM	Access method
OPEN	INPUT/OUTPUT/ EXTEND/REVERSE/ UPDATE/OUTIN/ INOUT/SINOUT	OPEN mode
RECFORM	FIXED LENGTH  VARIABLE LNGTH  UNDEFINED LNGTH	File consists of fixed-length records File consists of variable-length records File consists of records of undefined length
RECSIZE	nnnnn	Record length in bytes
BLKSIZE	nnnnn	Block length in bytes
BUFOFF	nnn L	Length of buffer displacement Buffer displacement according to file HDR2 label or default.
KEYPOS	nnnnn	Position of the ISAM key
KEYLEN	nnn	Length of the ISAM key in bytes
BLKCTRL	NO DATA	Data format without key Data format without key Block control information at beginning of block
POOLLNK	PAMKEY	Data format with PAM key
LOGLEN	aaaaaaaa nnn	ISAM pool link name Length of a logical flag in the ISAM index in bytes
VALLEN	nnn	Length of a value flag in the ISAM index in bytes
VALPROP	MIN  MAX	The lowest value flag within a data or index block is placed in the related index entry in the next higher level As above: the highest value flag is transferred
DUPEKY	YES NO	Duplicate ISAM keys may occur Duplicate ISAM keys are not permitted.
PAD	nnnnn	Padding factor
OVERLAP	YES	If a second I/O area is defined in the program, read operations can be executed in overlap mode
SHARUPD	NO YES	No overlapped processing File may be processed concurrently by more than one job
WROUT	NO YES  NO	No shared-update processing Every updated block is written back to disk immediately Updated block not written back to disk immediately

Output field	Values	Description
LABEL	NO	File labels are not processed
	STD	Standard labels, interchange level 1
TPMARK	(STD,n)	Standard labels, interchange level n
	NSTD	Tape file with non-standard labels
CODE	YES	Tape marks are written
	NO	No tape marks are written
	EBCDIC	No code conversion required
TRANS	ISO7	Conversion: EBCDIC <—> ISO7
	OWN	Code conversion using tables created by the user
BLIM	YES	ISO7 or OWN code converted to EBCDIC
	NO	ISO7 code converted to an 8-bit format by inserting a leading zero
CHKPT	nnnnnn	Up to nnnnnn data blocks per tape
	(NO,)	No automatic checkpointing
	(BLIM,)	Checkpoint written automatically on reaching block limit
	(FEOV,)	Checkpoint written automatically at every FEOV macro call
	(BLIM,)	Checkpoint written automatically on reaching block limit and at every FEOV macro call
FSEQ	(,DUMMY)	File is treated as a dummy file in the event of a restart
	(,ACTIVE)	File is processed further in the event of a restart
WRCHK	nnnnn	File sequence number in the file set
	UNK	Start position of the file is unknown
TAPEWR	NEW	New file at end of file set
	YES	Read-after-write check
CLOSMSG	NO	No read-after-write check
	DEVICE-BUFFER	Buffered output to magnetic tape cartridge
	IMMEDIATE	Immediate output to magnetic tape cartridge
	YES	Completion message after CLOSE
	NO	No completion message after CLOSE

FILE	Status information for the TFT entry linked with "pathname" is displayed. Otherwise entry selection is not based on the "filename".
=pathname	stands for:     [:catid:][\$userid.]filename
catid	Catalog ID of the pubset on which the file is stored. Default value: the catalog ID assigned to the user ID; (JOIN entry).
userid	User ID to which the file is assigned. Default value: user ID from the LOGON command
filename	Partially or fully-qualified file name (up to 41 characters). If "filename" denotes the file generation for a group, the absolute value of the generation number must be specified. When temporary files are specified, the internal file name is output.
LINK	
=link	Specifies the file link name.  If the LINK operand is specified, only the status information for the corresponding TFT entry is output. Otherwise the entry is not selected on the basis of the file link name.
=(BLANK,no)	Specifies the number of the TFT entry with the LINK name C'_____'.  The OPEN macro or the FILE command generates such entries if no LINK name or LINK=C'_____ ' is specified in the FCB (not recommended).  If the FILE and LINK operands were omitted, the status information for all TFT entries associated with the job is displayed.

## LINKAGE

Requests information on file linkage status. The output fields are explained below:

Output field	Values	Description
STATUS	ACTIVE INACTIVE	File being processed File not being processed
COMMAND	HOLD FILE	Link name locked with HOLD TFT entry created with FILE command or macro
DELAYED	OPEN RELEASE	TFT entry created with OPEN The RELEASE command issued for the link name was delayed due to a HOLD.
	RELEASE, KEEP	As above, when RELEASE was specified with KEEP
	RELEASE, UNLOAD	As above, when RELEASE was specified with UNLOAD
DATA (for tape files)	nnn	Number of tapes containing data of the file

## SECURITY

Requests information on file saving and retention periods. The output fields are explained below:

Output field	Values	Description
RETENT. PERIOD	NONE	No retention period in effect for the file
SECLEV	nnnnn	Retention period in days
	HIGH	Security level for label checking (for standard labels only)
	LOW	
OVWRITE PROTECT	YES	File protected against overwriting
BYPASS	NO	No overwrite protection
	NONE	No entry for label handling
	LP	No label handling; tape position remains unchanged
	(LP,nnnnn)	No label handling; tape is positioned to the specified tape mark, counting from the beginning of the tape
	(LP,-nnnnn)	No label handling; tape is moved forward the specified number of tape marks
	(LP,-nnnnn)	No label handling; tape is moved back the specified number of tape marks
DESTOC	NONE	No entry to destroy data till end of tape
	YES	After EOF/EOV labels have been written, the data on the remainder of the tape is erased
	NO	Data not erased till tape end

## VOLUMES

Requests information on volumes and devices. The output fields are explained below:

Output field	Values	Description
DEVICE	aaaaa	Device type
TSET NAME	NONE aaaa NONE	No entry for device type Name of tape set TFT entry not linked with tape set
TSET SHR	nnnnn	Number of TFT entries linked with tape set
FILE SET ID	aaaaaa	File set identifier
TSET VSN	(aaaaaa)	Volume serial numbers of tapes from the tape set (usually several values)
VSN/DEV	aaaaaa/aaaaaaaa NONE	VSNs and device type codes of volumes linked with the TFT entry (possibly several pairs of values) No entry

The device type code is only output if the device is requested or the volume (tape) is mounted.

The TSET attributes are not output if the value "NONE" is entered in the "TSET NAME" field. TSET SHR specifies the file number, which corresponds to the number of TFT entries linked with a TST entry.

FILE SET ID specifies the file set identifier (of the TST entry), whose value was ascertained using a FILE command command or at the time of OPEN.

TSET VSN corresponds to the device list of the TST entry. The volume serial number shown in parentheses is that of the current volume.

## Output format for TFT entries

The TFT entries are displayed sorted by file link names.

Depending on where, and on which volume, the file is located, the output format is as follows:

For files on public disk:

```
%                               LINK=linkname
                               FILE=filename
```

For files on private disk:

```
%D VOL#/DEV#                   LINK=linkname
                               FILE=filename
```

For files on tapes:

```
%T VOL#/DEV#                   LINK=linkname
                               FILE=filename
```

## Remote File Access (see also the "RFA" manual [12])

If the TFT entry for a file located within a remote system is to be displayed by means of RDTFT, the format for this display field is as follows:

```
%R                               LINK=linkname
                               FILE=filename
```

"R" means that the file concerned is remote. The catalog containing the file is specified within the pathname by means of the "catid".

**Example:**

- Status information on the TFT entries linked to the file TRICK is requested:

```
(IN)   /RDTFT FILE=TRICK
(OUT)  %      LINK=EDRPRIMR
        FILE=TRICK
(OUT)  %      LINK=IOOLINK
        FILE=TRICK
```

- Status information on the TFT entries linked with the file EDTISAM is requested, together with additional details on file security:

```
(IN)   /RDTFT LINK=EDTISAM, SECURITY
(OUT)  %      LINK=EDTISAM
        FILE=TICK
        RETENT. PERIOD =00015      SECLEV = HIGH
        OVWRITE PROTECT=NO
```

- Information concerning the HOLD status and the success of the RELEASE action is gathered using the LINKAGE operand:

```
(IN)   HOLD EDTISAM
(IN)   REL EDTISAM
(IN)   RDTFT LINK=EDTISAM, LINKAGE
(OUT)  %      LINK=EDTISAM
        FILE=TICK
        STATUS = INACTIVE HOLD
        COMMAND = FILE      DELAYED = RELEASE
```



## RELEASE Delete TFT entry

Application group: Device and volume reservation (page 29)

### Command description

The RELEASE command allows you to delete an entry in the task file table (TFT) by specifying the appropriate file link name. All tapes and tape devices linked to this entry, i.e. all those requested for the file are released. You can choose

- whether the devices remain allocated to the job (KEEP operand) or are returned to the system;
- whether the volume serial numbers (VSNs) of the private volumes are to remain known to the system, or be removed from it (UNLOAD operand). If the entry of the task file table refers to a TST (Task Set Table), the file number is decremented by 1.

If the file number decremented by 1 is 0, the TST entry and all the devices associated with the TST entry are released. If the file number decremented by 1 is greater than 0, then only those devices associated with the TST are released whose entries precede those of the specified device.

If the TFT entry does not refer to a TST, then all devices associated with that entry are released.

If there is more than one active file on a magnetic tape due to be released, this volume is not released until a RELEASE command has taken effect for each corresponding TFT entry.

The RELEASE command is ignored (even though e.g. the UNLOAD command may have been specified) if the associated TFT entry was previously locked by a HOLD command. The RELEASE command is not processed until the lock is removed by a DROP command or until the end of the job.

If a file was exclusively reserved with the FILE operand of a SECURE command, and subsequently processed, the RELEASE command ends this reservation.

### Format and operand description

Operation	Operands
$\left. \begin{array}{l} \text{RELEASE} \\ \text{REL} \end{array} \right\}$	link[, KEEP] [, UNLOAD]

- link                    Specifies the file link name of the TFT entry.  
If "link" is omitted, the first TFT entry with link name C'\_\_\_\_\_ ' is processed.
- KEEP                   Specifies that the tape devices linked with this file or TFT entry are not to be returned to the system, but are to remain available to the job for allocation purposes.
- UNLOAD                For tape files only.  
Specifies that the VSNs of the tapes linked with this file or TFT entry are to be removed from the system. This means that automatic reallocation of these volumes by the system is no longer possible (no AVR function).  
Tape devices will be unloaded.

**Example:**

Extract from a batch job:

```

/SECURE T9N=3 _____ (01)
/FILE PR.OUTPUT, LINK=IN, VOLUME=PR0001, DEVICE=T9N } _____ (02)
/FILE PR.OUTPUT, LINK=OUT, VOLUME=PR0002, DEVICE=T9N }
/FILE LINK=WORK, DEVICE=T9N, VOLUME=PR0003 }
/EXEC PR.ACCOUNTING
/RELEASE IN, UNLOAD _____ (03)
/RELEASE OUT _____ (04)
/RELEASE WORK, KEEP, UNLOAD _____ (05)

```

- (01) Three 9-track tape devices are reserved for this batch job.
- (02) Before the program PR.INVOICE is called, the requisite files are defined, i.e. the file link names IN, OUT and WORK within the program are linked with file characteristics.
- (03) The volume PR0001 is removed from the system and the tape device is returned to the system.
- (04) The tape PR0002 remains known to the system and the tape device is returned to the system.
- (05) The volume is removed from the system but the tape device remains allocated to the job.

## REMARK      Insert remark into command file

Application group: Job control (page 22 ff.)

### Command description

The REMARK statement allows you to insert remarks into command files (SYSCMD or procedure files).

The REMARK command can be issued any number of times.

If the REMARK command for a procedure file is to be logged to SYSOUT, then the operand "A" or "C" must be specified in the file's PROCEDURE command.

### Format and operand description

Operation	Operands
REMARK	remark

remark                      Any sequence of characters may be specified here; any entries in procedures extending beyond column 72 are truncated.

### Example

A procedure file starts with the following commands:

```

/PROCEDURE C, (&A,&B) _____ (01)
/REMARK PROCEDURE FOR LINKING _____ (02)
.....

```

(01) The entry "C" in the PROCEDURE command permits output of the remark to SYSOUT.

(02) The message "PROCEDURE FOR LINKING" is written to SYSOUT. In an interactive job this is displayed on the terminal in the following format:

```
%/REMARK PROCEDURE FOR LINKING
```

For further examples, see the SKIP and STEP commands.

## REMOVE-ISAM-POOL-LINK

### Remove ISAM pool link name from the table of pools

Application group: file processing (page 26 ff.)

#### Command description

The REMOVE-ISAM-POOL-LINK command can be used to remove one or all task-specific pool link names from the table of pools. The command is rejected as long as the file linked with the pool link name is open.

A user participating in an ISAM pool cannot terminate the link to this pool until all pool link names that are defined have been removed from the table. The same rule applies to the deletion of an ISAM pool.

#### Format and operand description

Operation	Operands
REMOVE-ISAM-POOL-LINK	LINK-NAME={ *ALL } [name]

LINK-NAME	Designates one or all task-specific pool link names to be removed.
=*ALL	All pool link names defined by the caller are removed from the table of pools - with the exception of those pool link names linked with a file that has not yet been closed.
=name	Pool link name to be removed.

## RESTART Restart program from checkpoint

Application group: Program control (page 30)

### Command description

The RESTART command causes the program to restart at a point (checkpoint) defined using the WRCPT macro, see the "Executive Macros" manual [5]. The program is loaded and executed starting at the specified point. The resources associated with the program at the time of the WRCPT macro are assigned and positioned to the appropriate block.

This command can be used both in interactive and batch jobs.

The task initiating the restart requests the same memory as the interrupted task had when the checkpoint was taken. In addition, the task initiating restart determines whether the restarted task will operate in interactive or batch mode. Files which were open at the time when the program and operating system status were saved are also open when restart takes place. However, EAM files are not reconstructed.

If these files are file generations, care must be taken to ensure as far as possible that the base value does not change in relation to the WRCPT time. The RESTART command ignores any update of the file generation group between the macro and the RESTART command and adopts the status at WRCPT time.

The RESTART command may only be used if the SYSDTA system file is combined with SYSCMD and if the assignment of system files SYSIPT, SYSLST and SYSOUT is not a primary assignment (PRIMARY operand).

The WRCPT macro must have been issued in the same version of BS2000 and in the same system configuration as the RESTART command, otherwise you will receive an error message. No new SDF-SYNTAX file may be assigned between WRCPT and RESTART.

Before issuing the RESTART command, the tapes which were processed when the checkpoint was encountered have to be remounted. The checkpoint data contains all information the system requires in order to reposition the tapes to the appropriate block.

In the event of a shortage of class 5 memory space, RESTART processing is terminated with an error message.

The life span of a temporary file extends at the most from LOGON to LOGOFF. You must therefore either reconstruct the temporary files prior to the RESTART command or declare them as dummy files for the restart (FCB, FILE : CHKPT=DUMMY or RESTART : DUMMY= ).

The copy of a procedure/ENTER file created by the operating system (S.IN.tsn. ....) is

not (automatically) deleted when LOGOFF is entered, if checkpoints were set during the job run. This file must be in existence if restart is to be problem-free.

CHECKPOINT/RESTART cannot be used to extend the life span of a temporary file.

Following RESTART, access can only be made to temporary files of one's own task. (Following RESTART, it is not possible to access temporary files of the WRCPT task unless the WRCPT task and the RESTART task are identical.)

If, because of insufficient space, the SYSLST system file assignment has been changed from disk to tape between the time the WRCPT macro was executed and the RESTART command was issued, an initialized disk file must again be made available at the time RESTART is entered.

Any allocation existing between the STXIT routine and the "SVC event class" - made at the time the WRCPT macro is executed - is cancelled after the RESTART command and must be reset by means of the STXIT macro (after the WRCPT macro).

If RESTART command processing aborts with the message "EXC0305 PAM I/O-ERROR (xx)", this may be due to the following reasons:

xx	Meaning
X'04'	REQM error
X'08'	Catalog error (e.g. file non-existent)
X'0C'	A device cannot be used.
X'10'	A file is opened in shared-update (SHARUPD) mode.
X'14'	No extension for slot segment available or the slots could not be re-linked.
X'18'	Response "T" to MSG DDEE, DDED or error when writing to checkpoint file
X'1C'	Non-matching VSNs or error with ISAM reopen
X'20'	Error in FCB
X'24'	The number of tape devices for a file is smaller than that at the time the checkpoint was set.
X'2C'	Error on reopening a SAM file (logicals).
X'40'	Checkpoint function not supported.
X'44'	It is not possible to call CHKPT and write checkpoints at the tape end of VLTFs (Very Large Tape Files), since the C12 option EREPASSW=Y has been set.

The status indicator in the monitoring job variable is set to "\$R" at RESTART time (see the "Job Variables" manual [11]).


If "jvname" is not accessible at command execution time, an error message is issued to SYSOUT and processing aborts.

RESTART processing causes existing RFA connections to be cleared down.



## Format and operand description

Operation	Operands
RESTART	<p>pathname1 [, page] [, LOAD]</p> <p>[ , CHECK={<math>\left. \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \right\}</math>}]</p> <p>[ , IDENT=name]</p> <p>[ , VSEQ={<math>\left. \begin{array}{c} \text{filesectno} \\ \text{LAST} \end{array} \right\}</math>}]</p> <p>[ , CHKPT=number]</p> <p>[ , DUMMY={<math>\left. \begin{array}{c} \text{pathname2} \\ (\text{pathname2}, \dots) \end{array} \right\}</math>}]</p> <p>[ , MONJV=jvname]</p> <p>[ , TEMPJV={<math>\left. \begin{array}{c} \text{YES} \\ \text{NO} \end{array} \right\}</math>}]</p>

pathname1	stands for: [:catid:][userid.]filename1
catid	Catalog ID of the pubset on which the file is stored. Default value: the catalog ID assigned to the user ID (JOIN) entry.
userid	User ID to which the file is assigned Default value: user ID from the LOGON command.
filename1	Name of the file created by the WRCPT macro and containing the program to be loaded.
page	Number of the PAM block where the checkpoint records begin. They are created by the WRCPT macro, and this number is output to SYSOUT. From these checkpoint records, the system obtains information for the restart.

- LOAD** Once the program is loaded, it is not to be started; the system goes into command mode (see also the LOAD command).
- CHECK**  
**=YES** Checks whether the internal file names of the files to be opened at RESTART have changed. If an internal file name has changed, the restart is aborted.
- =NO** Any change in the internal file names will be ignored.
-  **CHECK=NO** must be specified if checkpoint files have been migrated by HSMS in the meantime.
- CHKPT=number** Number of checkpoints to be executed in the recovery of defective file sections.  
Valid entry:  $0 \leq \text{number} \leq 255$
- This operand allows the user to resume processing from a given checkpoint and terminate when the required number of checkpoints have been written. The identifiers of the checkpoints are reinserted.  
Default value: number = 0.
- DUMMY** Indicates one or more files which are not required for restart. The checkpoint file can only be specified if the CHKPT=... operand is not specified.
- =pathname2** stands for [:catid:][userid.]filename2. For "catid", "userid", see above. "filename2" is the name of file.
- =(...)** List with file names. Up to 255 file names may be specified.
- IDENT=name** A 6-byte identifier which identifies the checkpoint. In checkpoint processing (WRCPT), this identifier is displayed on SYSOUT together with a number (see "page" operand).
- If two checkpoints of identical identifiers exist for a given checkpoint file, then IDENT refers to the last set checkpoint. The previously set checkpoint can then only be referenced by the "page" operand.  
If neither "page" nor IDENT is specified, the program restarts from the last checkpoint.
- The IDENT operand is only effective if the checkpoint file is a disk file.



MONJV=jvname	<p>Name of a job variable with which the restarted program can be monitored. The value of this job variable is set by the operating system and depends on the processing state of the restarted program.</p> <p>Possible values for the monitor job variable: \$R, \$T or \$A (see the "Job Variables" manual [11]).</p> <p> The MONJV operand must not be used together with the LOAD operand.</p> <p> The MONJV operand is only available with the JV software product.</p>
TEMPJV	<p>Specifies that the names of temporary job variables are to be checked</p> <p>=YES and processed. If an error occurs in this regard, the restart is aborted.</p> <p>=NO The names of temporary job variables are neither checked nor processed.</p>
VSEQ=filesectno	<p>Indicates a file section number where the restart is to begin.</p> <p><math>0 \leq \text{sectno} \leq 255</math>.</p> <p>This operand may only be used for files with standard labels, cataloged with FSEQ=1 (see the FILE command).</p> <p>The operands "page", LOAD and IDENT must not be specified together with VSEQ.</p> <p>Default value: sectno = 0 (ordinary restart).</p> <p>=LAST Indicates the last file section.</p>

**Examples:***Example 1:*

The third tape of a tape sequence is destroyed. Reconstruction by means of the RESTART command from the preceding checkpoint.

```
/RESTART TAPE.SAVE, VSEQ=2, CHKPT=1
```

The program terminates when the first checkpoint is written. No catalog update is performed.

*Example 2:*

The checkpoint with the identifier FP001 from the checkpoint file CP.FILE1 is to be started. Any change of internal file name following the WRCPT call is to be ignored. The file TEST1 is to be treated as a dummy file following restart.

```
/RESTART CP.FILE1, ID=FPT001, CHECK=NO, DUMMY=TEST1
```

## RESUME      Switch from system to program mode

Application group: Program control (page 30)

### Command description

The RESUME command permits a loaded program to be started or, after an interruption, to be resumed at the point where it was interrupted. The format shown below is incomplete.

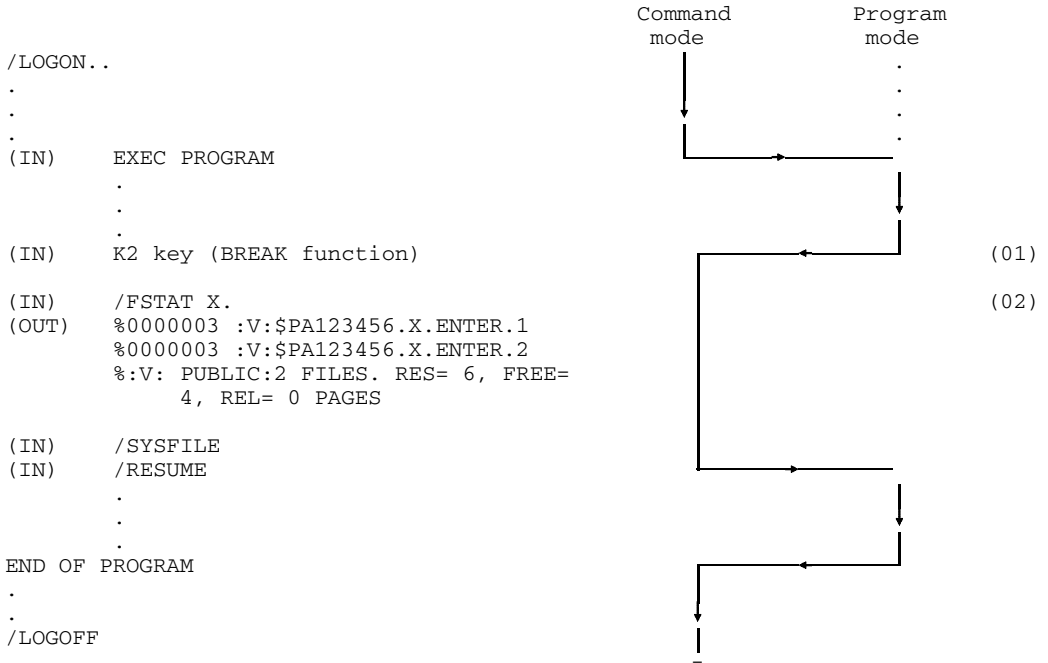
The RESUME command is an IDA (Interactive Debugging Aid) command. The "Interactive Debugging Aid" manual [4] describes the expanded form of this command, which allows resumption of a program at any given point.

### Format and operand description

Operation	Operands
$\left. \begin{array}{l} \{ \text{RESUME} \} \\ \{ \text{R} \} \end{array} \right\}$	

**Example:**

At any time during interactive mode operation, a program interrupt can be effected using the ESCAPE or BREAK function, i.e. by pressing the appropriate key on the data terminal. The program remains loaded, and the system waits for commands to be entered on the terminal.



- (01) Program execution is interrupted by the BREAK function and a branch made at the same time to system mode. The RESUME command causes a return to the program at the point where it was interrupted.
- (02) The FSTATUS command can be entered, without affecting the subsequent return to the loaded program by means of the RESUME command. This applies to all commands with the exception of the EXECUTE, LOAD, RESTART or RFAEND command.

## RFAEND Clear down RFA connection

Application group: Remote File Access, RFA (page 37)

The RFAEND is only provided with the RFA software product (see also "RFA" manual [12]).

### Command description

The RFAEND command terminates the AFR partner job and clears down the RFA connection.

If no RFASTART command was issued previously, this command is rejected.

In cases where one or more catalog IDs and an RFASTART command were issued in the job without a corresponding RFAEND command, an implicit RFAEND command is executed for all the RFASTART commands at LOGOFF time.

If you have entered a number of RFASTART commands for the same catalog ID, you must issue the same number of RFAEND commands for that catalog ID in order to terminate the corresponding AFR partner job (application: procedure nesting; see CALL command).

– The RFAEND command is illegal for a currently loaded program.

### Format and operand description

Operation	Operands
RFAEND	catid

**catid** Specifies the catalog ID or the remote system on which the RFA partner job is to terminate. If "catid" is a catalog ID of the user's own system (local processor), the command will be formally accepted.

For examples, refer to the "RFA" manual [12].

## RFASTART      Set up RFA connection

Application group: Remote File Access, RFA (page 37)

The RFASTART command is only available with RFA software product (see also the "RFA" manual [12]).

### Command description

The RFASTART command allows you to establish an RFA connection, at the same time opening an RFA system run and generating an AFR partner job in the remote system whose catalog ID is "catid".

Every processor on which an RFASTART command is to be given must have an appropriate entry of "catid" with the processor name in MRSCAT before /RFASTART is executed. This entry can be generated by the command "/CATM catid,HOST=bcamname". The processor name can also be added subsequently (prior to /RFASTART) by setting up an MRS connection.

Only one AFR partner job is generated on the remote system even if more than one RFASTART command is given for that remote system.

If there has been a successful RFASTART command for a given catalog ID, another RFASTART command is accepted without error message (for procedure nesting).

### Format and operand description

Operation	Operands
<pre>{ RFASTART } { RFAS      }</pre>	<pre>catid [ ,USER=string] [ ,ACCOUNT=string] [ ,PASSWORD=string] [ ,TIME={ value }         { IMMED } ]</pre>

- catid Catalog ID of the system to which an RFA connection is to be established.
- In the case of one-character catalog IDs, "catid" is the 4th character of the volume serial number (VSN) for public volumes (PUBxzz; x=catid). In the case of catalog IDs comprising 2-4 characters, "catid" is the part to the left of the period (e.g. CDE,01; CDE=catid).
- If the catalog ID is not entered in the MRSCAT table, the entry is invalid, or the catalog inaccessible, the command is rejected.
- The information whether the catalog ID has been entered in the MRSCAT table can be obtained by means of the STAM command.
- If "catid" refers to the local system (system executing the command, home or imported catalog), then no AFR partner job is generated. However, the command is formally accepted.

ACCOUNT=string

Specifies the account number for the generated AFR partner job. The convention of the LOGON command applies. The account number of the RFA job is used by default.

PASSWORD=string

Specifies the LOGON password which refers to the generated AFR partner job. The LOGON command convention applies. No password is assumed if the PASSWORD operand is omitted.

*Notes:*

- The PASSWORD operand must be specified in the /RFASTART command even if the password of the AFR partner job is the same as that of the RFA job.
- User ID, account number, and password are also verified by the remote system.
- The other LOGON information entries (job name and MSG operand) for the AFR partner job are obtained for the RFA job.

TIME	Specifies a period in seconds.
=value	$0 \leq \text{value} \leq 99999999$  The system, when unable to set up connection to the remote system at the time of the RFASTART command, tries to establish a connection to the remote system every 32 seconds within the specified period (for TIME=32, one retry). If no connection can be set up within that period, you will receive an error message.  However, RFASTART processing is not reattempted if any of the following errors occur: <ul style="list-style-type: none"><li>– Invalid RFASTART operand</li><li>– USERID, ACCOUNT#, PASSWORD were not accepted by the remote system referenced by the catalog ID, "catid".</li></ul>
= <u>IMMED</u>	You will receive an error message immediately after an unsuccessful attempt to set up an RFA connection.
USER=string	User ID for the generated AFR partner job. The convention of the LOGON command applies. The identifier of the RFA job is used by default.

For examples, refer to the "RFA" manual [12].



## RJOB Assign name to batch job

Application group: Device control (page 33)

The RJOB command will not be supported in versions following BS2000 V10.0A.

### Command description

The RJOB command allows a name to be given to a batch job and enables the user to control the output from the job (spoolout). Output may be directed to a batch terminal or to a printer at the computer center.

At a batch terminal (the currently valid RLOGON command contains your user ID) the user can elect to have the output printed once the job is complete or can defer it until an output request is initiated by an ROUT command. You may also specify an additional user ID authorizing another user to receive the output. In this case, the output is only produced once, i.e. at the time at which it is first requested; in other words, the first prospective recipient requesting output will be served.

The RJOB command is optional. If it is omitted when the job is input at a batch terminal, then the system enters a job name (RBP.tsn); output is produced immediately after the job is completed.

The RJOB command always relates to the job which immediately follows it, i.e. it must precede the associated LOGON command.

### Format and operand description

Operation	Operands
RJOB	jobname [, {CENTRAL DEFER[,userid]}]

- jobname** This name identifies the job within the system. The name enables you as user at the batch terminal to specify both the timing and the location of the output from the job (ROUT command), and to obtain information on the progress of the job (RSTATUS command). A job which has the same name as a job which already exists in the system will be rejected.  
The job name consists of up to 8 alphanumeric characters and special characters.
- CENTRAL** This specifies that output from the job is to be directed to a printer in the computer center immediately after job processing.
- DEFER** The output from the job is to be retained in the system until it is requested by an ROUT command. This operand applies only to output to be printed at batch terminals and cannot be used for a local printer.
- DEFER,userid** In this case, the output is to be deferred and an additional batch terminal user, whose user ID is specified in "userid", is authorized to receive the output. If this user is currently active, (s)he is informed of job execution.  
Output from a job can only be printed once; it is received by the first authorized user who issues the output request (via a ROUT command).



If the CENTRAL and DEFER operands are not specified in the RJOB command, then the output is printed immediately following completion of the job to the batch terminal from which the job was entered. If this batch terminal is inactive at that time, the output is retained by the system until the terminal is reactivated or the user logs onto the system at a different batch terminal by issuing an RLOGON command. In this case, the operator is informed that output is available for an inactive batch terminal.

**Examples:***Example 1:*

Input and output at the same batch terminal:

Fig. RJOB-1 Input/output at the batch terminal

Input on the card reader at the batch terminal:

```

.....
/RLOGON ABC, ACCNR
/RJOB RDEMO1
/LOGON ABC, ABT140UD, C'UD'
.....
} ----- (01)
/LOGOFF
}
}
}
/RJOB RDEMO3, DEFER
/LOGON XYZ, ABT140FD, C'B34'
.....
} ----- (02)
/LOGOFF
}
}
}
/LOGON XYZ, ABT140FD, C'B34'
.....
/LOGOFF
/STATUS J=RDEMO3
/ROUT J=RDEMO3
.....
} ----- (03)
}
}
}

```

- (01) The RJOB command assigns the name RDEMO1 to the following job. This name is used in all the logging messages relating to the job. The termination of the spoolin process is reported on the batch terminal printer with the following message (for TSN 0978 at 17:19 hrs):

```
0978.1719 R002 JOB ACCEPTED          RDEMO1      ABC
```

The following message indicates that execution of job RDEMO1 is complete ("ABC" is the user ID in the RLOGON command):

```
0978.1721 R003 JOB COMPLETE          RDEMO1      ABC
```

Output from the job follows this last message.

- (02) This job is assigned the name RDEMO3. The logging messages correspond to those for job RDEMO1. Because DEFER is specified in the RJOB command the output from the job is deferred by the system. On completion of the job, the output can be obtained by issuing an ROUT command.
- (03) There is no RJOB for this job.

Its name is formed from "RBP" and the job number (TSN). The log message reporting job completion could be, for example:

```
0982.1724 R003 JOB COMPLETE          RBP.0982    ABC
```

This message is then followed by the output from the job.

*Example 2:*

The job is spooled in from a batch terminal, but output from the job is directed to a printer at the computer center:

Fig. RJOB-2 Input from the batch terminal

- 1) The following cards are among those read in on the batch terminal card reader:

```
/RJOB JOBNO4,CENTRAL  
/LOGON ABC,DEPT140UD,C'UD'  
.....  
/LOGOFF
```

} job JOBNO4

- 2) Once execution of the job is complete, the output is produced on the printer as soon as this is available.

*Example 3:*

The RJOB command may also be used in local batch processing in order to assign the output to a batch terminal:

Fig. RJOB-3 Output to the batch terminal

- 1) The following job is entered at the computer center:

```
/RJOB JOBNO5,DEFER,ABC  
/LOGON ABC,DEPT140UD,C'UD'  
.....  
/LOGOFF
```

} job  
JOBNO5

- 2) The use of the DEFER operand causes the output from job JOBNO5 to be directed to the batch terminal where output is requested by user ABC using an ROUT command:

```
/RLOGON ABC,ACCNO  
/ROUT J=JOBNO5
```

For further examples, see the ROUT, RMSG and RSTATUS commands.

## RLOGOFF      Terminate batch terminal input

Application group: Device control (page 33)

The RLOGOFF command will not be supported in versions following BS2000 V10.

### Command description

By using the RLOGOFF command you indicate that you have completed your input at a batch terminal.

The user will then have three options for subsequent action:

1. The batch terminal can remain linked to the system in order to receive output.
2. You can specify a new RLOGON command, making further input possible.
3. An RSTOP command disconnects the batch terminal from the system.

Even where no RLOGOFF command is entered, there are two cases where the system performs RLOGOFF processing: an RLOGON command is in force but you enter another RLOGON command, or you may enter an RSTOP command.

### Format

Operation	Operands
RLOGOFF	

*Example:*

The following cards are input at the batch terminal:

```
/RSTART STATION1
/RLOGON XYZ, ACCNR
/LOGON XYZ, ABT140FD, C' B34 '
.....
/LOGOFF
```

} job

More jobs are input as follows:

```
/RSTATUS U=XYZ
/RLOGON ABC, ACCNR _____ (01)
/RSTATUS U=ABC
/RLOGOFF _____ (02)
/RSTOP
```

(01) The system effects an RLOGOFF and issues the following log message to the batch terminal printer:

```
1078.1636 R005 RLOGOFF EFFECTED BY THE SYSTEM
```

(02) The following log message is output:

```
1078.1643 R001 COMMAND ACCEPTED /RLOGOFF
```

For a further example see the LOGON command.



## RLOGON      Initiate batch terminal input

Application group: Device control (page 33)

The RLOGON command will not be supported in versions following BS2000 V10.

### Command description

The RLOGON command enables jobs to be input from a batch terminal. To do so, a valid user ID and account number must be entered in the command. This defines the current user of the batch terminal, i.e. the person authorized to enter remote batch processing commands such as RJOB, ROUT, RMSG and RSTATUS which refer explicitly to the user (operand U=userid), and to receive output for jobs entered subsequently.

The RLOGON command remains in force until

- a new RLOGON command,
- an RLOGOFF command or
- an RSTOP command is entered.

You must not log on concurrently at more than one batch terminal using the same user ID. If you wish to transfer to a different batch terminal, you must first terminate your session at the original terminal (RLOGOFF) before issuing the RLOGON command at the other terminal.

### Format and operand description

Operation	Operands
RLOGON	userid,accountno[,password]

userid	User ID, account number and password (as given in the LOGON command). Further operands of the LOGON command may be specified, but they will be ignored.
accountno	
password	

## Example

Card deck input on the card reader at the batch terminal:

```
/RSTART STATION1 _____ (01)
/RLOGON ABC, ACCNR _____ (02)
/LOGON ABC, ABT140UD, C'UD'
..... }
/LOGOFF } 1st job

/LOGON XYZ, ABT140FD, C'B34'
..... }
/LOGOFF } 2nd job

/LOGON ABC, ABT140FD, C'UD'
..... }
/LOGOFF } 3rd job
/RLOGOFF
/RSTOP
```

- (01) These two cards are mandatory; without them the following jobs would not be accepted by the operating system.
- (02) The following logging message is output to the batch terminal printer to register the acceptance of the RLOGON command:

```
1053.1507 R001 COMMAND ACCEPTED /RLOGON ABC
```

This command identifies ABC as the user of batch terminal STATION1 and as the owner of the jobs that follow.

For further examples, see the RLOGOFF, RSTATUS, RJOB and ROUT commands.

## RMSG Issue message from batch terminal

Application group: Device control (page 33)

The RMSG command will not be supported in versions following BS2000 V10.0.

### Command description

By using the RMSG command, batch terminal users can issue a message to

- another batch terminal user,
- a batch terminal connected to the system or
- the control console in the computer center.

The system rejects messages destined for terminals which are not connected to the system or directed to users who are not currently active. (A user is regarded as active if either his user ID was specified in an RLOGON command which is still in force, or if, after his RLOGOFF command, no new RLOGON command has been entered at the terminal which is still connected to the system.) If you specify that a message is to be sent both to a user and to a batch terminal, then the system attempts to send the message to the user first. Only if the specified user is not active does the system try to send the message to the specified batch terminal.

### Format and operand description

Operation	Operands
RMSG	M=C'text' [,U=userid] [,T=termid]

**M=C'text'** Message text: It is enclosed in single quotes and must not be longer than 40 characters, including blanks.

**T=termid** Terminal identifier: This specifies the batch terminal to which the message is to be sent. If the terminal is not linked to the system, then the message is rejected. If the terminal ID is specified as "BS2.CTR", then the message is displayed on the main console.

**U=userid** User ID: This specifies the user who is to receive the message. If this user is not active, then the message is rejected. When the user ID "BS2.OPR" is specified, the message is displayed on the control console.

*Example:*

Input on the card reader at batch terminal STATION1:

```
.....  
/RLOGON XYZ,ACCNO  
/RMSG M=C'PLEASE ASSIGN PRINTER 3',T=BS2.CTR _____ (01)  
/RJOB JOBNO1,CENTRAL _____ (02)  
/LOGON...
```

(01) The following logging message is printed on the batch terminal printer:

```
1080.1705 R001 COMMAND ACCEPTED  
/RMSG M=C'PLEASE ASSIGN PRINTER 3',T=BS2.CTR
```

At the computer center, the following message is displayed on the console:

```
%1080.170550 %C R008 0 1705 FROM : XY STATION1 TO : BS2.CTR  
MESSAGE: PLEASE ASSIGN PRINTER 3
```

(02) The output for the subsequent job JOBNO1 is to be printed on a printer in the computer center (CENTRAL operand in the RJOB command).

## ROUT Request job output

Application group: Device control (page 33)

The ROUT command will not be supported in versions following BS2000 V10.

### Command description

The ROUT command enables batch terminal users to request deferred job output.

The command has no effect if

- the job is not present in the system;
- job execution is not yet complete (the command must be entered again after reaching the end of the job);
- the user is not authorized to receive the output, i.e. he has neither entered the job, nor has he been named as an additional recipient of the output in the associated RJOB command.

In all cases an appropriate message is issued.

You can use the ROUT command to withdraw jobs from the system so that no output is received. You can only remove jobs which you have submitted yourself and which are currently in the system. If these conditions are not fulfilled, you will receive an appropriate message from the system. Jobs can be removed any time, once they have been transferred.

If the output from a job has already been transmitted, the ROUT command is meaningless, since all information on the job will have been deleted.

### Format and operand description

Operation	Operands
ROUT	$\left[ \left[ \left\{ \begin{array}{l} J=\text{jobname} \\ U=\text{userid} \end{array} \right\} [, \left\{ \begin{array}{l} \text{CONTINUE} \\ \text{BEGIN} \\ \text{DELETE} \end{array} \right\} ] \right] \right]$ $\left[ U=*ALL [, \left\{ \begin{array}{l} \text{CONTINUE} \\ \text{BEGIN} \end{array} \right\} ] \right]$

BEGIN	Output commences with the first record even if it has been interrupted.
<u>CONTINUE</u>	Output of the job is to continue, commencing in front of the record at which the output was interrupted.
DELETE	<p>This operand has the following effect:</p> <ol style="list-style-type: none"><li>1. The job (J=jobname) specified in the command is deleted from the system. The job must belong to the current user, i.e. his RLOGON command at the batch terminal must be currently in force.</li><li>2. All jobs present in the system which have been submitted by the user (U=userid) are deleted. The entry "U=userid,DELETE" is only accepted from the user whose RLOGON command is currently in force. The system reports the names of all the jobs which have been deleted.</li></ol> <p>If both operands (J=jobname, U=userid) are omitted, the system continues an interrupted output or releases the device to receive further output.</p>
J=jobname	This specifies that the request relates to a particular job, whose name has been defined in an RJOB command.
U=userid	Causes the request to apply to the specified user ID. If you specify your own user ID, you receive, or remove (by using DELETE), all output from jobs which have been submitted. If you specify the user ID of a different user, then you receive all pending outputs from that user's jobs where you have been declared as an authorized recipient in the RJOB or PRINT command.
U=*ALL	You request output on all jobs for which you as the user are the authorized recipient. This applies both to jobs which you yourself have submitted and to other users' jobs for which you have been named as an authorized recipient in the RJOB or PRINT command.

*Example*

Input on the card reader at the batch terminal:

.....

```

/RLOGON XYZ,ACCNO
/RJOB JOBNO01,DEFER
/LOGON XYZ,DEPT140FD,C'B34'
..... } job JOBNO01
/LOGOFF
/RSTATUS U=XYZ _____ (01)
/ROUT U=XYZ _____ (02)
.....

```

- (01) The RSTATUS command is used to determine if there are any outputs pending for the specified user ID "XYZ" ("OUTPUT AVAILABLE").
- (02) The following logging message is issued on the batch terminal printer:

```
0978.1928 R001 COMMAND ACCEPTED /ROUT U=XYZ
```

Following this message, the outputs for user ID "XYZ" are printed out, e.g. for job JOBNO1 whose output has been deferred until now because the RJOB command specified DEFER.

If the output is not yet available, the ROUT command is rejected with the message "JOB EXECUTING" and it must be input again later.

For a further example, see the RJOB command.

## **RSTART      Activate batch terminal**

Application group: Device control (page 33)

The RSTART command will not be supported in versions following BS2000 V10.

### **Command description**

The RSTART command is used to activate a currently inactive batch terminal.

A batch terminal is regarded as inactive if it is physically connected to the computer but has not been initiated by the entry of an RSTART command.

Fig. RSTART-1      Activate batch terminal

Once the batch terminal has been activated by the entry of the RSTART command, it can receive output and the user can log on to the operating system (see the RLOGON command).

The RSTART command includes a specification of a name for the batch terminal.



## Format and operand description

Operation	Operands
RSTART	termid [,FORM=code]

termid	<p>Terminal identifier.</p> <p>This specifies the name of the batch terminal as defined by the system administrator. The name consists of up to 8 alphanumeric characters of which the first must be alphabetic. If a terminal with this name is already connected to the system, then the RSTART command is rejected ("R050 REMOTE STATION NOT IDENTIFIED").</p>
FORM=code	<p>Specifies the stationery to be used for output. The form number "code" consists of 1 to 6 alphanumeric characters and must have been defined in consultation with the system administrator. If this operand is omitted or FORM="STD___" is specified, standard forms will be used.</p> <p>Only those PRINT requests can be output at the batch terminal whose FORM entries match the FORM option of the RSTART command.</p>

### *Example:*

Input on the card reader at the batch terminal:

```
/RSTART STATION2
```

Logging output to the batch terminal printer:

```
1053.1507 R001 COMMAND ACCEPTED /RSTART STATION2
```

This activates the terminal.

For further examples, see the RLOGON, RLOGOFF and RSTATUS commands.

## RSTATUS Request status of remote batch jobs

Application groups:

Device control (page 33)

Interrogation of current commands (page 31)

The RSTATUS command will not be supported in versions following BS2000 V10.

### Command description

The RSTATUS command enables the user of a batch terminal to interrogate the status of remote batch jobs present in the system at the time of command processing.

You can request status information on

- a particular job,
- the jobs related to a particular user ID, or
- all jobs input from a particular batch terminal.

The system provides this information only to authorized recipients. These are users under whose RLOGON command the job was input or who are named as authorized recipients in the job's RJOB command.

After issuing the RSTATUS command, you receive the following message:

```
%R006      job:  userid,device,recipient,priority,tsn,form status
```

The message components are explained below:

job	Job name
userid	Job initiator, i.e. the user ID of the RLOGON command which was valid at the time of job entry.
device	Name of the batch terminal from which the job was entered.
recipient	User ID of another recipient, provided he was named in the RJOB command.
priority	Priority of the job.
tsn	TSN of the job.
form	Form number as specified in the RSTART command.

status	Status of a job. Any of the following may be displayed:	
	EXECUTING:	Job executing (TYPE 2)
	AWAITING:	Job in wait state (TYPE 1)
	OUTPUT AVAILABLE:	Job processed
	ABNORMAL TERMINATION:	Self-explanatory
	RESERVED BY SUBMITTER:	Job processed; output reserved for the job initiator
	RESERVED BY ALTERNATE:	Job processed; output reserved for the recipient.
	NONE:	Job not in system (if operand "U" or "T" is specified)
	%R034	COMMAND REJECTED, JOB NOT IN SYSTEM Job not in system (if operand "J" is specified)
	%R035	COMMAND REJECTED, INVALID RECIPIENT Illegal request; user not authorized.

### Format and operand description

Operation	Operands
RSTATUS	$\left\{ \begin{array}{l} J=\text{jobname} \\ U=\text{userid} \\ T=\text{termid} \end{array} \right\}$

J=jobname	The command applies to the specified job name. If you are not authorized to request a display of the status of this job, the command is rejected.
T=termid	The status of all jobs currently in the system which have been input from this terminal is requested. It does not matter whose jobs they are, i.e. the user's authorization is irrelevant.
U=userid	The status of all the jobs of the user with the specified user ID is reported, provided the user has the authority to request it. If you specify your own user ID then you are informed of the status of all remote batch jobs currently in the system.

*Example:*

Input on the card reader at the batch terminal:

```
/RSTART STATION2
.....

/RLOGON XYZ,ACCNO
/RJOB RDEMO2,DEFER,ABC
/LOGON XYZ,ABT140FD,C'B34'
.....
} job
/RLOGOFF
/RSTATUS J=RDEMO2 _____ (02)
```

(01) \_\_\_\_\_

(01) The output for job RDEMO2 is to be deferred until it is expressly requested (see the ROUT command).

(02) Sometime later, the status of deferred job RDEMO2 is requested. The following logging message is output on the batch terminal printer:

```
1070.1701 R001 COMMAND ACCEPTED /RSTATUS J=RDEMO2
1070.1701 R006 JOB : RDEMO2 XYZ STATION2 ABC
STATUS : NORMAL TERMINATION OUTPUT AVAILABLE
```

*Note:*

Output is assigned a standard job name or the job name "SOUTtsn".

For further examples, see the RJOB, ROUT and RLOGOFF commands.

## RSTOP Deactivate batch terminal

Application group: Device control (page 33)

The RSTOP command will not be supported in versions following BS2000 V10.

The RSTOP command enables you to terminate the connection set up by the RSTART command between a batch terminal and the operating system.

Before the connection is severed, all messages intended for output at this terminal are transmitted. Once the last of these messages has been output, no more input or output at this terminal is allowed. Operation can only be resumed by entering an RSTART command.

If the batch terminal is connected to the system via a dial-up line, then the connection is cleared down.

### Format

Operation	Operands
RSTOP	

### *Example:*

Input on the card reader at the batch terminal:

```
/RSTOP
```

Logging message output to the batch terminal printer:

```
1078.1643 R001 COMMAND ACCEPTED /RSTOP
```

This renders the terminal inactive. If a dial-up has been used, the connection will be cleared down.

For further examples, see the RLOGON and RLOGOFF commands.

## RTI Return to interrupted procedure

Application group: Job control (page 22 ff.)

### Command description

The RTI command enables you to leave ESCAPE mode (ESCAPE command) and return to the interrupted procedure.

If a program was loaded at a procedure level lower than that at which the interrupt took place, then it is taken into the system file area where the RTI command (MODE=PROGRAM) originated (see example).

Setting the RTI command is mandatory for resumption of an interrupted procedure; in other words, no other command performs the function of an RTI command.

Logically, the RTI command must always be entered from the terminal, so that the user can leave the ESCAPE mode. If, however, it occurs within a procedure or in the primary SYSCMD of a batch job, an error results, and the SPIN-OFF mechanism is activated.

If an interrupt status is to be reset by one procedure level (via an ENDP or DO command; see examples 1 and 2 of ESCAPE command), then the specification for the interrupt mode (i.e. PROGRAM or COMMAND) is ignored. If this new level was already in the interrupt mode, it retains its interrupt mode; if not, the interrupt mode is COMMAND.

### Format and operand description

Operation	Operands
RTI	[MODE={ COMMAND PROGRAM}]

**MODE=COMMAND**The procedure is to be continued in the command mode. In the event of an interruption in the program mode, the program remains loaded until a further RESUME or RTI command (MODE=PROGRAM) occurs.

MODE=PROGRAMThe procedure is to be continued in the program mode. If no program is loaded, an error message is output, and control returns to the terminal. You can then enter the appropriate command (/RTI MODE=COMMAND).



If the MODE operand is omitted, the procedure continues in the mode prevailing at the moment of interrupt.

Command mode: Control returns to the next command in the procedure file.

Program mode: The interrupted program resumes at the address at which the interrupt occurred.

### Example

Fig. RTI-1 Program execution in different areas

## SYSCMD assignments

- (1) SYSCMD is assigned to the procedure file PROC1 by means of the command `"/CALL PROC1"`.
- (2) The procedure PROC1 is interrupted in the program mode (i.e. the interrupt takes place during execution of the program PROG). SYSCMD is assigned to the data display terminal.
- (3) SYSCMD is assigned to the procedure file PROC2 by means of the command `"/CALL PROC2"`.
- (4) The procedure PROC2 is interrupted in the command mode via an ESCAPE command. SYSCMD is now assigned to the data display terminal.
- (5) The program PROG is initially stored in the system file area of the procedure PROC2. After the program run, control is passed to the procedure file PROC2. Processing continues from the command which follows the ESCAPE command.
- (6) SYSCMD is once more assigned to the data display terminal (ESCAPE mode still effective at level 1). There follows a branch from procedure level 2 to procedure level 1.
- (7) The RTI command cancels the ESCAPE mode. Since at this point the program PROG is no longer loaded, the `"MODE=COMMAND"` operand must be specified. If this operand is not entered, an error message is issued. (Procedure PROC1 was interrupted in the program mode; however, there is no longer a program loaded.) The complete command must then be entered.
- (8) SYSCMD is once more assigned to the primary command input.



## SECURE Request resources

Application groups:     File processing (page 26)  
                           Device and volume reservation (page 29)

### Command description

The SECURE command can be used in both interactive and batch modes in order to request or release resources that are required by a job while it is executing.

The SECURE command is only supported for reasons of compatibility. Use of the SECURE-RESOURCE-ALLOCATION command is recommended instead.

### Format

Operation	Operand
<pre>{ SECURE } { SEC }</pre>	<pre>[ , T9P=n ] [ , T9G=n ] [ , TAPE-C1=n ] [ , TAPE-C2=n ] [ , disk device type=n [ , ... ] ] [ , UNIT= (mn, ... ) ] [ , VOL= (vsn/type, ... ) ] [ , WORK=n ] [ , FILE= (pathname [ /EX ] , ... ) ]</pre>

**Operand description (in alphabetical order).**

**FILE** Implicitly reserves all private volumes, together with the associated devices, on which the specified file is located. Disks and their devices are considered shareable (system private volumes). If two or more file names are specified in the FILE operand, they must be separated from each other by commas, and the whole group must be enclosed in parentheses. A maximum of 48 entries is supported.



If the FILE operand refers to a file generation group, no volumes or devices will be reserved.

**=(pathname)** "pathname" stands for [ :catid: ][ \$userid. ]filename

**catid** Catalog ID of the pubset on which the file is stored.  
Default value: the catalog ID assigned to the user ID (JOIN entry).

**userid** User ID to which the file is assigned.  
Default value: user ID from the LOGON command.

**filename** Fully-qualified name of a cataloged file or name of a file generation (file generation groups may only be reserved exclusively). Both system and user files may be reserved; the system merely checks whether the file or file generation exists. If not, the job is not inserted in the device queue; instead, an error message is issued. In batch mode, control is passed to the next STEP, ENDP, ABORT, ABEND or LOGOFF command.

**=(pathname[/EX],...)**

Specifies exclusive reservation of a file, file generation or file generation group, i.e. in addition to any existing implicit reservation of volumes and devices, a file lock is effected (LOCK) which prevents any other job from accessing the file(s).

If "filename" is the name of a file generation, the appropriate reservation is performed; the associated file generation group is locked.

If "filename" is a file generation group, the whole group is locked against foreign access, and the lock can only be removed via a new SECURE command or via the LOGOFF command. This reservation of a file generation group is to be recommended whenever file generations are added to the said group. This protects the group from another job simultaneously changing its base value (see the BASE operand in the CATALOG command).

If one of the requested resources is not available, further processing depends on whether it is interactive or batch mode:

Interactive mode:

The command is rejected and a corresponding message is issued.

Batch mode:

The job is put into a job queue (the SECURE queue) with unlimited maximum wait time until all requested resources are available at the same time.

Files cataloged under the user ID of the system administrator can only be reserved exclusively in the case of jobs running under this ID. Files currently in use when the reservation request is made, however, can no longer be reserved. In this case, in batch processing, the requesting job is put in the device queue, and in interactive processing, a corresponding message is issued.

TAPE-C1=n	Specifies the number of magnetic tape devices (18 track format 3480) to be requested, if any.
TAPE-C2=n	Specifies the number of magnetic tape devices (18 track format 3480 XF) to be requested, if any.
T9P=n	Specifies how many ("n") magnetic tape devices that operate in the PE (1600 bpi) recording method are to be reserved.

T9G=n

This can be used to request "n" GCR 9-track magnetic tape devices (6250 bpi) for the job.

At SECURE time, these device allocations are not yet assigned to a permanent UNIT. The system merely ensures that the required number of tape devices supporting the corresponding recording density are kept available and cannot be allocated to other jobs (unimodal and bimodal tape devices).

A UNIT assignment occurs:

- with /FILE file,VOL=...,DEV=...,LINK=link
- when the tape file is OPENed.

A UNIT assignment is cancelled:

- with the return of the device allocation with SECURE or RELEASE following CLOSE,
- after RELEASE link,KEEP, which retains the device allocation without a UNIT assignment.

A tape device reserved for each device type can be used for both DMS and SPECIAL applications. Volumes can be remounted.

UNIT=(mn,...)

Specifies that the device with the mnemonic device name "mn" (2 or 4 bytes) is to be reserved. This reservation is exclusive. Up to 32 devices may be specified.



If a tape device is involved, the reserved device may be used for DMS applications as well as applications in the usage mode SPECIAL. By contrast, disk devices that are reserved with the UNIT operand may only be operated in the usage mode SPECIAL. However, since the usage mode SPECIAL is only available from privileged applications, a disk device reserved with UNIT is blocked to normal users for any kind of use.

The volume that is mounted on the UNIT may not be remounted on any another device. This means that processing will be aborted if an unrecoverable hardware error occurs on the device in question.

It is therefore preferable to assign a tape or disk device via a corresponding device type (explicit device assignment) or via a VOLUME or corresponding file reservation (implicit device assignment).

VOL=(vsn/type,...)

Specifies the 6-byte volume serial number (VSN) and the volume "type" for the volume to be secured. This also implicitly reserves the required device. In this instance, mass storage disks and their associated disk storage unit are considered shareable, i.e. the volume implicitly becomes a shareable private disk.

The entries in the VOL operands must be enclosed in parentheses and separated by commas. A maximum of 16 entries is allowed. The following device types may be specified:

Disk storage units:

D3475  
D3480  
D348E  
D3436  
D3437  
D348F  
D3435  
D3439-10

Magnetic tape devices:

T9P  
T9G

Magnetic tape cartridges:

TAPE-C1  
TAPE-C2

WORK=n

Specifies the number of tape terminals to be reserved for the task. The recording density of the assigned device is arbitrary, but the device can only be used for processing work files (DEVICE=WORK in the FILE command). Here, too, a UNIT assignment only takes place with "FILE ...,LINK=...,DEV=WORK" or with OPEN.

disk-device-type

Specifies the type of disk device to be requested. The disk device types listed in the device table in the appendix can be specified.

*Exception:*

D3490-xx and devices with device type code C4 may only be requested with the SECURE-RESOURCE-ALLOCATION command.

=n

Number of disk devices for each device type.

The following table provides an overview of the reservations brought about by the various operands in the SECURE command:

Operand in SECURE command	Resources reserved		
	Files	Private volumes	Devices for private volumes
Type of disk device	-	-	Disk devices of the type specified are reserved exclusively (available in usage mode SPECIAL only)
FILE=(pathname, ...)	Files and file generations on private disks are reserved as shareable.	All associated private disks are reserved as shareable (implicit assignment).	Requisite disk devices are reserved as shareable (implicit assignment).
FILE=(pathname [/EX],...)	Tape files are reserved exclusively.	The tape on which the file start is reserved exclusively (implicit assignment).	Requisite tape devices are reserved as shareable (implicit assignment).
	Files (also file generations or file generation groups) are reserved exclusively. <sup>1)</sup>	All associated private disks are reserved as shareable (implicit assignment).  No reservation takes place for a file generation group.	Requisite disk devices are reserved as shareable (implicit assignment).
	Files (also file generations or file generation groups) on private disks are reserved exclusively. <sup>1)</sup>	-	-
T9P, T9G, TAPE-C1, TAPE-C2	-	-	Tape devices of the type specified are reserved exclusively.

Operand in SECURE command	Resources reserved		
	Files	Private volumes	Devices for private volumes
UNIT= (mn,...)	-	-	Devices of any type (except consoles) are reserved exclusively.
VOL= (vsn/type, ...)	-	Disks with the specified volume serial numbers are reserved as share- able; tapes are re- served exclusively.	Requisite disk devices are reserved as share- able; tape devices are reserved exclusively (implicit assignment).
WORK=n	-	Tapes selected by the operator are reserved exclusively.	Requisite tape devices are reserved exclu- sively (implicit assignment).

- 1) The individual file is locked against foreign access; in the case of file generations and file generation groups, this extends to the entire associated file generation group.

## **SECURE-RESOURCE-ALLOCATION      Request resources**

Application groups:

- File processing (page 26 ff.)
- Device and volume reservation (page 29)

### **Command description**

The **SECURE-RESOURCE-ALLOCATION** command enables the user to reserve resources required by a job for its execution. This reservation guarantees that the system will not reject later attempts to access resources. Resources include:

- devices for private volumes (disk devices, tape devices),
- private volumes (disks, tapes),
- files on private and public volumes.

Reserving a file also causes the disk or tape to be reserved, and with them the corresponding devices. For information about the device types specified here please refer to the device table in the appendix.

Tape/data volume and device reservations are always exclusive.

A resource is reserved exclusively if no other job may use it as long as it is reserved. It is shareable if the system - although ensuring that the resource is freely accessible during the time of reservation allows other jobs to use it.

Explicit reservation of public volumes is rejected.

You may use the **SECURE-RESOURCE-ALLOCATION** command in batch and interactive mode.

When you issue the **SECURE-RESOURCE-ALLOCATION** command, you release all previously reserved private resources (implicit **RELEASE**). TFT entries are not released and therefore new ones do not have to be created. However, the device assignments linked with the TFT entries are cancelled.

An existing reservation can be cancelled as follows:

- repeating the **SECURE-RESOURCE-ALLOCATION** command
- **RELEASE** command:
  - RELEASE KEEP** cancels the reservation of the file and, if applicable, of the associated disk or tape. Device reservations are retained, however.
  - If disks/tapes or devices which are assigned to the file are also implicitly reserved by other files or disks/tapes, the former are not released until all claims on them by the system have been eliminated.
- Reentering the **SECURE-RESOURCE-ALLOCATION** command
- **WHEN** command
  - Same function as the **SECURE-RESOURCE-ALLOCATION** command without



operands.

- LOGOFF command (end of job)

Other users' files can only be reserved if they are entered in the catalog with SHARE=YES (see CATALOG command). Files under the system administrator's ID can only be reserved exclusively by jobs executing under this ID.

If a volume is requested via the FILE operand in the SECURE-RESOURCE-ALLOCATION command, this is designated an implicit request. If the same device type is requested both explicitly and implicitly with one command, the number of devices assigned is the sum of the explicit and implicit device requests. The device assignments no longer conflict.

### *Example*

```
/SECURE-RESOURCE-ALLOCATION DEV=(TYPE=T1600,N=5),TAPE=(VOL=A104BC,TYPE=T1600)
```

Result: 6 devices are reserved.

If the SECURE-RESOURCE-ALLOCATION command is used to reserve a volume which has not yet been mounted, a MOUNT message is issued at the console.

If the operator rejects a request for a resource (MOUNT message at the console), no reservation is made at all. In batch mode, the system branches to the next STEP or LOGOFF command.

If a file that takes up a number of private volumes is reserved, the reservation always includes all of these volumes.

If one of the requested resources is unavailable, no reservation is made at all. Depending on the type of job and application, the WAIT operand specified (only applies to SECURE-RESOURCE-ALLOCATION command), the system either issues an appropriate message or the job waits in the SECURE queue until all the requested resources are available to the job - or until the defined wait time expires. However, if the implicitly requested devices do not exist or the required number of devices is not available, the system branches to the next STEP or LOGOFF command.

If the SECURE-RESOURCE-ALLOCATION command cannot be executed owing to lack of resources, volumes previously reserved for this job are released.

The system rejects the SECURE-RESOURCE-ALLOCATION command if the calling job (or job making the request) has opened files on private volumes that are to be exclusively reserved. Exception: When the SECURE-RESOURCE-ALLOCATION command originates from a procedure file which is itself on private volumes, or a file on private volumes was previously assigned using the SYSDFILE command.

Files migrated to background storage by HSMS (see "HSMS" manual [21]) can be

returned to normal memory by means of `SECURE-RESOURCE-ALLOCATION, FILE=filename` (implicit recall). In this case, HSMS outputs a **RECALL SUMMARY** report.

**Remote File Access** (for details refer to "RFA" manual [12]):

The `SECURE-RESOURCE-ALLOCATION` cannot reserve local and remote volumes simultaneously.

The local TFT contains entries for all remote files being processed. Exclusive reservations of remote files and remote resources are ignored after the `/RELEASE` command has been issued.

Files in a "REMOTE" system can also be reserved via RFA. However, only files may be specified in this command and they must all belong to the same system.

If an RFA partner task is executing on a system version  $\leq$  Version 7, the `SECURE-RESOURCE-ALLOCATION` command must be issued.

### Reserve (disk) files, disks and disk devices

Whether or not a disk may be reserved depends on the values specified for the "USER-ALLOCATION", "SYSTEM-ALLOCATION" and "OPERATION-CONTROL" operands in the "SET-DISK-DEFAULTS" and "SET-DISK-PARAMETER" operator commands. These values can be displayed by means of the SHOW-DISK-STATUS command or by using an appropriate NKDINF macro call.

Shareable private disk (SPD):

A SECURE-RESOURCE-ALLOCATION FILE=... for files located on an SPD is indicated in the F1 label of the private disk. Other processors accessing this SPD respect this reservation.

If a pubset specified by the "catid" entry in the FILE operand is not locally available and not in the RFA table of the job, command execution is terminated with an error message. It is therefore not possible to wait for local availability of a pubset.

For SPECIAL applications (e.g. VOLIN, FDDRL), disk devices can also be reserved via the UNIT and DEVICE operands. However, special privileges are required in order to use devices reserved in this way (see also the SECURE command).

### Format

Operation	Operands
$\left. \begin{array}{l} \text{SECURE-} \\ \text{RESOURCE-} \\ \text{ALLOCATION} \\ \text{SEC-RES} \end{array} \right\}$	$\left[ \text{DISK} = \left\{ \begin{array}{l} *NO \\ (\text{VOLUME}=\text{vsn}, \text{TYPE}=\text{device} \\ \\ \text{[ , ALLOCATION} = \left\{ \begin{array}{l} \text{EXCLUSIVE} \\ \text{SHARED} \end{array} \right\} ] ) L \end{array} \right\} \right]$ $\left[ \text{ , FILE} = \left\{ \begin{array}{l} *NO \\ (\text{NAME}=\text{pathname} [ \text{ , ALLOCATION} = \left\{ \begin{array}{l} \text{SHARED} \\ \text{EXCLUSIVE} \end{array} \right\} ] ) \right\} \right]$ $\left[ \text{ , WAIT} = ( \text{[ TIME} = \left\{ \begin{array}{l} \text{TASK-STD} \\ n \end{array} \right\} ] [ \text{ , EVENT} = \left\{ \begin{array}{l} \text{ALL-MOUNT} \\ \text{DISK-MOUNT} \end{array} \right\} ] ) \right]$

L#<VR100>Indicates operand expressions that can be specified in list form (op1,op2,...).#1

#<BS>L#<BE.Vr100>Example:#1

#<BS>L#<BE.VR100>FILE=((NAME=file1,MOUNT=2),(NAME=file2,MOUNT=1))

**Operand description**

DISK={...}

- \*NO No reservation of disk and disk device; default value; -
- VOLUME=vsn Volume serial number of the private disk to be reserved.
- TYPE=device Specifies the disk device type on which the disk is to be mounted. For permissible disk device types, see device table in the appendix.
- ALLOCATION Specifies the reservation mode for the disk.
- =EXCLUSIVEThe disk cannot be used simultaneously by other users (exclusive reservation).
- =SHARED The disk can be used simultaneously by other users (shareable); default value.

FILE={...}

- \*NO No reservation of a file; default value.
- NAME=pathname specifies the file to be reserved.
- pathname stands for: [:catid:][userid.]filename
- catid Catalog ID of the pubset on which the file is stored. Default value: catalog ID assigned to the user ID (JOIN entry).
- userid User ID to which the file is assigned. Default value: user ID from the LOGON command.
- filename Fully-qualified name of a cataloged file, file generation or file generation group. System or user files may be reserved; the system merely checks whether the file or file generation exists. If not, the job is not put into the device queue; instead, an error message is issued. In batch mode, a branch is made to the next STEP, ABORT, ABEND command or to the LOGOFF command. All disks associated with the file are reserved and mounted. The disks are always shareable.
- ALLOCATION Specifies the reservation mode of the file.
- =SHARED Both the file and the corresponding disks or devices are reserved as shareable. The file may be used by other users (both read and write access allowed), i.e. only the disks and devices required implicitly will be reserved. SHARED is the default value.

=EXCLUSIVE The file must not be used simultaneously by other users. The reservation of the corresponding disks and devices remains shareable. It is only possible to reserve a file exclusively if it is not being used. If a file generation group is involved, the entire file generation is protected against unauthorized access.

WAIT={...}

TIME Specifies the maximum length of time that the job will wait for processing of the reservation request.

=TASK-STD A distinction is made between interactive and batch mode:

Interactive mode:

The command is rejected if the reservation requests cannot be satisfied immediately. In other words, the job does not wait for required resources to be released. However, it does wait until any requested disk is mounted and a corresponding confirmation is received from the operator.

Batch mode:

The wait period is unlimited. Within the prescribed time period, the job will wait indefinitely for resources to be mounted or released. If the requests have not been fulfilled after the wait period has elapsed, a branch is made to the next STEP, ENDP or LOGOFF command.

=n Waiting time in seconds; accuracy lies in the minute range. The maximum value is 43200 seconds (=12 hrs). If less than 180 seconds is specified, the command is rejected if a disk needs to be mounted or if operator confirmation is required (in contrast to TASK-STD).

### Example

A batch job reserved a file X.ADVANCE with the command:

```
/SECURE-RESOURCE-ALLOCATION FILE=(NAME=X.ADVANCE, ALLOCATION=EXCLUSIVE)
```

whereupon an interactive job trying to read the file receives the error message "LOCK ERROR ....".

**Reserve (tape) files, tapes and tape devices**

The SECURE-RESOURCE-ALLOCATION is rejected if

- a) a program is loaded and USE=SPECIAL assignments exist for the task
- b) a tape file is open
- c) a tape file is in the HOLD status.

The recording density specified by the SECURE-RESOURCE-ALLOCATION command operands must match the density with which the tape is to be processed, otherwise other tape devices will be assigned in addition to those reserved by SECURE-RESOURCE-ALLOCATION.

If a file extending over a number of tapes is reserved, the SECURE-RESOURCE-ALLOCATION command reserves only one device (for the first volume). The number of tape devices to be reserved may be specified via the MOUNT operand of SECURE-RESOURCE-ALLOCATION.

A tape volume reserved via /SECURE-RESOURCE-ALLOCATION VOLUME=... (explicit reservation) or SECURE-RESOURCE-ALLOCATION FILE=... (implicit reservation) has a VOLUME-PHASE equal to PREMOUNT (see the SHOW-TAPE-STATUS and SHOW-RESOURCE-ALLOCATION commands). In other words, this tape is merely reserved in advance to prevent the same volume from being used by another job. However, I/O operations do not yet take place for the volume. An implicit device reservation exists. Between OPEN and CLOSE, the tape has a PHASE equal to IN-USE (the tape is being processed). Following CLOSE, the PHASE again returns to PREMOUNT.

## Format

Operation	Operands
$\left. \begin{array}{l} \text{SECURE-} \\ \text{RESOURCE-} \\ \text{ALLOCATION} \\ \text{SEC-RES} \end{array} \right\}$	$\left[ , \text{DEVICE} = \left\{ \begin{array}{l} \text{*NO} \\ \text{(TYPE=device, NUMBER=n)} \end{array} \right\} \right]^L ]$ $\left[ , \text{UNIT} = \left\{ \begin{array}{l} \text{*NO} \\ \text{mn, ...} \end{array} \right\} \right]$ $\left[ , \text{TAPE} = \left\{ \begin{array}{l} \text{*NO} \\ \text{(VOLUME=vsn, TYPE=device} \\ \quad \left[ , \text{ACCESS} = \left\{ \begin{array}{l} \text{READ} \\ \text{WRITE} \end{array} \right\} \right] \left[ , \text{MOUNT} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \end{array} \right\} \right]^L ]$ $\left[ , \text{FILE} = \left\{ \begin{array}{l} \text{*NO} \\ \text{(NAME=pathname [ , ALLOCATION} = \left\{ \begin{array}{l} \text{SHARED} \\ \text{EXCLUSIVE} \end{array} \right\} \right] \\ \quad \left[ , \text{ACCESS} = \left\{ \begin{array}{l} \text{READ} \\ \text{WRITE} \end{array} \right\} \right] \left[ , \text{MOUNT} = \text{n} \right] \end{array} \right\} \right]^L ]$ $\left[ , \text{WAIT} = \left( \left[ \text{TIME} = \left\{ \begin{array}{l} \text{TASK-STD} \\ \text{n} \end{array} \right\} \right] \left[ , \text{EVENT} = \left\{ \begin{array}{l} \text{ALL-MOUNT} \\ \text{DISK-MOUNT} \end{array} \right\} \right] \right) \right]$

L#<VR100>Indicates operand expressions which can be specified in list form (op1,op2,...).#

#<BS>L#<BE.VR100>Example:#

#<BS>L#<BE.VR100>FILE=((NAME=file1,MOUNT=2),(NAME=file2,MOUNT=1))

**Operand description**

DEVICE={...}

TYPE=device Specifies the type of magnetic tape device. The following types can be specified:

Device	Characteristics
TA	Magnetic tape device of unspecified recording density
T1600	Magnetic tape device with recording density of 1600 bpi
T6250	Magnetic tape device with recording density of 6250 bpi
BM16...	Bimodal tape device (see appendix A.1)
TAPE-C1	Magnetic tape cartridge, 18 track, uncompressed
TAPE-C2	Magnetic tape cartridge, 18 track, compressed

NUMBER=n Number of devices to be reserved;  
default value: n = 1.

UNIT=mn Mnemonic device name (2 or 4 characters) of the tape device to be reserved. Reservation is always exclusive. Apart from a reservation via the device type, this is the only way to reserve resources for volume processing without using DMS functions.



TAPE={...}

VOLUME=vsn Specifies the volume serial number (VSN) of the magnetic tape to be reserved.

TYPE=device Specifies the type of magnetic tape device. The following types can be specified:

Device	Characteristics
TA	Magnetic tape device of unspecified recording density
T1600	Magnetic tape device with recording density of 1600 bpi
T6250	Magnetic tape device with recording density of 6250 bpi
BM16...	Bimodal tape device (see appendix A.1)
TAPE-C1	Magnetic tape cartridge, 18 track, uncompressed
TAPE-C2	Magnetic tape cartridge, 18 track, compressed

ACCESS Informs the operator whether a write-enable ring is required for the tape.

=READ The tape is to be read only. No write-enable ring is required. This operand is used in the premount message to inform the operator whether the tape is to be mounted with or without a write-enable ring. Thus, there is no write protection for tapes.

=WRITE The tape is to be written.  
The operator is requested in the premount message to mount a write-enable ring.

MOUNT Indicates whether a device is to be implicitly assigned to the tape due to be reserved and whether a premount message is output.

=YES In addition to tape reservation, device reservation also takes place. This operand also causes a premount message to be issued at the console.

=NO Only tape reservation is performed, no device reservation (offline reservation). No premount message is issued.

FILE={...}

NAME=pathname

"pathname" stands for: [:catid:][\$userid.]filename

catid Identifier of the catalog where the tape file is cataloged. Default value: catalog ID assigned to the user ID (JOIN entry).

userid User ID to which the file is assigned.  
Default value: user ID from the LOGON command.

filename Fully-qualified name of a cataloged file, file generation or file generation group. System or user files may be reserved; the system merely checks whether the file or file generation exists. If not, the job is not put into the device queue; instead, an error message is issued. In batch mode, a branch is made to the next STEP, ABORT, ABEND command or to the LOGOFF command.  
All tapes in the catalog are reserved. Device reservations and tape premounting occur, in accordance with the MOUNT operand. The value in the EVENT operand determines whether waiting takes place.

ACCESS Informs the operator whether or not a write-enable ring is required for the tape.

=READ The tape is to be read only. A write-enable ring is not required. This operand is used in the MOUNT message to inform the operator whether the tape is to be mounted with or without a write-enable ring. Thus, there is no write protection for tapes.

=WRITE The tape is to be written. The operator is requested in the mount message to mount a write-enable ring.

MOUNT=n Specifies the number of devices to be implicitly reserved, where  $0 \leq n \leq 55$ . Default value: 1.

"n" tapes are mounted on the "n" devices in the sequence of their catalog entry and MOUNT messages are output. The maximum number of devices implicitly reserved corresponds to the VSNs (for tapes) in your catalog entry. If "n" is less than the number of VSNs, the remaining tapes are reserved without implicit device reservation.

**ALLOCATION** This operand has no effect on tape files.

Reserving a file on private volumes causes the volumes entered in its catalog to be implicitly reserved. Since access to tape volumes is always sequential, a tape cannot be processed simultaneously by more than one job. Thus, by definition, tape volume reservation (whether implicit or explicit) is always job-specific. This in turn means that all files located on the tape volumes reserved by a job cannot be reserved/processed by another job until the reserving job releases the tape volume. This is equivalent to an explicit reservation of all files located on the reserved tape.

**WAIT={...}**

**TIME** Specifies the maximum amount of time that the job will wait for the reservation request.

**=TASK-STD** Identifies the default values. A distinction is made between interactive and batch mode:

Interactive mode:

The command is rejected if the reservation requests cannot be satisfied. Mount requests for tapes are handled as specified in the **EVENT** operand (see below).

Batch mode:

Within the prescribed time limits, the job will wait indefinitely for resources to be mounted and released. If the requests have not been honored by the time the wait period has elapsed, a branch is made to the next **STEP**, **ENDP** or **LOGOFF**.

**=n** Specifies the waiting time in seconds. Accuracy lies in the minute range. The maximum value is 43200 seconds (=12 hrs). If less than 180 seconds is specified, the command is rejected if a tape has to be mounted.

**EVENT** Specifies whether the user wishes to wait for the mount requests to be answered.

**=ALL-MOUNT**

The system waits for the operator to respond to all mount requests (mount requests occur synchronously).

**=DISK-MOUNT**

The system will not wait for a response to tape mount requests (it only waits for requested disks to be mounted). As a result, mount requests are synchronous for disks but asynchronous for tapes.

*Example*

```
/LOGON  
...  
/SECURE-RESOURCE-ALLOCATION TAPE=(VOLUME=(B0000A,TYPE=T1600) _____ (01)  
/EXEC PROG  
...  
/SECURE-RESOURCE-ALLOCATION _____ (02)
```

- (01) This SECURE-RESOURCE-ALLOCATION command exclusively reserves for this interactive job a tape with the volume serial number B0000A. A 9-track tape device is assigned to the job for this purpose.
- (02) All resources reserved for this job are released.

## SETJV Set job variable

Application group: Job variable functions (page 36)

The SETJV command is only available with the JV software product (see the "Job Variables" manual [11]).

### Command description

The SETJV command is used to assign a value to a user job variable.

Before this command may be entered for a job variable, the latter must be cataloged by means of a CATJV or DCLJV command.

Command processing terminates with an error message if the job variable specified in the "value" operand does not exist, has no value or if a substring specified with "jvname,start,length" does not exist.

If the actual length of the "value" field does not match the length entry in the operand "length", then the value denoted by "value" is truncated on the right or padded with blanks (X'40), as required, in accordance with "length". The maximum length (256 bytes) is, of course, not exceeded.

### Format and operand description

Operation	Operands
SETJV	$\left\{ \begin{array}{l} \text{jvid} \\ (\text{jvid}[, [\text{start}][, \text{length}]]) \end{array} \right\}, \text{value}$ [, PASS=password]

jvid	This entry may contain any of the following job variable names:
jvname	Specifies the fully-qualified name of a permanent or temporary job variable.
*jvlink	Specifies a job variable link name.
start	Specifies the first position to be changed in the job variable. If this entry is omitted, position 1 is assumed. The start position must be between 1 and 256.
length	Specifies the number of characters to be set. If specified, "length" must be between 1 and 256, where "start" and "length" together must not exceed 257.

If "length" is omitted, the entire job variable value, beginning at the position specified in "start", is replaced by the string specified in the operand "value". Thus, the length of the previous job variable value can be decreased, increased, or left unchanged.

If a length other than zero is supplied and the substring to be modified is within the old job variable value, its length remains unchanged. If a length other than zero is supplied and the substring to be modified or added exceeds the old job variable value, then the job variable value is lengthened.

value

With this entry the user specifies the value of the job variable, using one of the following formats:

1. Direct specification:

The value can be specified in either character format or hexadecimal format:

C'.....' or X'.....'

The maximum length of the constant defined by "value" is 253 characters (or 127 characters for hexadecimal entries, since only 256 characters are permitted for each operand in every command).

2. Adopting values from an existing job variable:

"value" may be:

- the name of a permanent job variable
- the name of a temporary job variable
- a job variable link name
- a special job variable name.

In this case, the job variable to be modified is set to the entire value of the cited job variable.

If "value" is entered in the form:

(jvname,start,length), the job variable to be modified is set to the substring, indicated by "start" and "length", from the value for job variable "jvname".

PASS=password

Indicates the read or write password with which the job variable is protected. PASS may be omitted if the password was specified earlier with the PASSWORD command.

For examples refer to the "Job Variable" manual [11].

## SETSW      Read or reset job switches

Application group: Job control (page 22 ff.)

### Command description

The SETSW command allows you to set the 32 job switches allocated to a job to ON or OFF, and to invert or read the current settings. Information about switch settings is obtained by issuing the command without operands.

The operating system provides 32 job switches for each job. The job switches are numbered 0 through 31 and stored in the TCB. Each switch can be set to ON or OFF or inverted separately. At the beginning of a job, the switches are set to OFF (unlike the user switches). It is up to you to define the meaning of the various switch settings for your job. When doing this, bear in mind that the job switches are used by some system components and utility routines as well. Switch settings are not retained beyond the end of the job (LOGOFF). Switches 16 - 31 are switched to OFF during execution of the STEP command.

### Format and operand description

Operation	Operands
SETSW	$[\text{ON}=\left\{\begin{array}{l} \text{no} \\ (\text{no}, \dots) \end{array}\right\}]$ $[, \text{OFF}=\left\{\begin{array}{l} \text{no} \\ (\text{no}, \dots) \end{array}\right\}]$ $[, \text{INVERT}=\left\{\begin{array}{l} \text{no} \\ (\text{no}, \dots) \end{array}\right\}]$

no	Number of job switch to be reset. There are 32 job switches, numbered 0 through 31.
ON=no (no,...)	The specified switches are set to ON.
OFF=no (no,...)	The specified switches are set to OFF.
INVERT=no (no,...)	The specified switches are set to ON if they were OFF, and vice versa.

*Example:*

```

/LOGON
...
(IN)   SETSW _____ (01)
(OUT)  % ALL SWITCHES SET OFF
(IN)   SETSW ON=(5) _____ (02)
(IN)   SETSW
(OUT)  % SWITCHES ON EQUAL-
(OUT)  % 5
(IN)   SETSW ON=(6,20),OFF=(5) _____ (03)
(IN)   SETSW
(OUT)  % SWITCHES ON EQUAL-
(OUT)  % 6,20

```

- (01) At the beginning of the job all switches are set to OFF.
- (02) Switch 5 is set to ON. The switch positions are subsequently read by means of the SETSW command (without operands).
- (03) Switches 6 and 20 are set to ON; switch 5 is set to OFF.

For further examples, see the PROCEDURE and STEP commands.



## Use of job switches in BS2000

When using job switches, bear in mind that some system components and software products alter the status of certain job switches or are controlled by them. The following table shows which system components or software products use job switches on a standard basis:

System component/Software product	Switches
ARCHIVE	30, 31
BCAMDEF	0, 4, 5, 31
DBL/ELDE (linkage editor/loader)	4
EDOR	1-8
EDT	4-7
FMS	5
LMS	1, 4, 8, 9
PRSERVE	0, 2, 3
STEP	16-31
SYSGEN/UGEN	1
TSOSLNK	4
MSGEDIT/MSGLIB	1

### In general

setting switch 4 suppresses message "%BLS0500" of the loader.

### ARCHIVE

The software product ARCHIVE can be called in DO as well as in ENTER procedures. Information about execution can be obtained from the switch positions set to ON or OFF by ARCHIVE during or after the run.

Switch 30 set to ON by ARCHIVE: Warning message in procedures.

Switch 30 is set to ON by ARCHIVE if the ARCHIVE statement was executed although a warning message was issued.

Switch 31 set to ON by ARCHIVE: Error in procedures.

Switch 31 is set to ON by ARCHIVE if the ARCHIVE statement was executed although an error was detected.

### **BCAMDEF**

Use of switches 0, 4, 5, 31:

Switches 0, 4, 5 and 31 are set to ON during the BCAMDEF procedure and then set to OFF.

(0, 31: cf. Assembler/4, 5: cf. EDT)

### **DLL/ELDE** (linkage editor/loader)

Switch 4 set to ON: system messages concerning the loading of a module (BLS0500, BLS0517, ...) are suppressed.

### **EDOR**

Switch 1 set to ON for EDOR: Alternative input source for commands/records

In interactive mode, EDOR normally reads the commands and data records from the screen via system function WRTRD. However, in certain cases (e.g. when calling DO procedures), it is desirable to read the data from another, previously prepared input source such as a file. This requires the use of two system commands: the logical system file SYSDTA is switched over by

```
/SYSFILE SYSDTA=filename
```

and job switch 1 is set to ON by means of the system command

```
/SETSW ON=1
```

EDOR will then read commands and/or data records from the specified input source via system function RDATA.

When a command error occurs, the screen is automatically returned to for this single input operation in conjunction with function WRTRD. The user can thus correct the error directly on the screen and have the command read in again without having to redefine the input source. Reading is then continued from SYSDTA via function RDATA.

The window display after each command sequence executed is suppressed as long as job switch 1 remains set to ON. Data records are therefore only expected on the screen during insertion; they are processed as in batch mode.

Character strings requested via the input variable are always read from the terminal and not from SYSDTA, provided that the terminal has been defined as the input medium (default option).

Thus the interactive nature of a search program, for example for parameter input, is preserved in this mode of operation.

Reading from the alternative input source continues until either job switch 1 is switched to OFF or the end of the file reached. In the latter case, EDOR sets job switch 1 to OFF and outputs the comment:

```
"xxSYSDTA=INTERACTIVE TERMINAL, PLEASE ENTER COMMAND xx"
```

on the screen. Subsequently, a command is expected as usual.

Switch 2 set to ON for EDOR: trace mode for search programs

When testing complicated search programs, it is often useful to know which statements have been carried out during the search program run. A trace mode can be activated by setting job switch 2 to on (ON=2). In this mode, a message is listed on the output medium defined in EDOR, for each statement with branch (program and/or data branch) which was carried out successfully. The messages have the format:

```
TRACE: ANWSG XXX/MARKE XX ("statement xxx / tag xx")
```

or, if the statement has no tag:

```
TRACE: ANWSG XXX ("statement xxx")
```

*Note:*

Trace mode can be activated even if a program loop has already been started, or is believed to have started, by proceeding as follows:

interrupt the program, set job switch 2 to ON and restart with /INTR.

EDOR will then output a question on how to continue, which must be answered with "+". Following this, all branches are listed as described above.

Switch 3 set to ON for EDOR: timing of command execution

Job switch 3, when set to ON, causes the CPU time (in seconds) required for the execution of a command or a command sequence to be displayed on the screen in the last positions of the commentary, when the latter appears after reading in the command or sequence. If there was no commentary, one is generated with the text "TIMING".

*Example:*

```
"xxxxxxx SEARCH TERMINATED. 2208 HITSxxx001.321"
```

Switch 4 set to ON for EDOR: input listing in batch mode

In batch mode, the output messages which are written to the system output medium SYSOUT as a standard procedure comprise only those generated by EDOR and possibly the output lines generated by \$- or S-commands. It may sometimes be useful for checking purposes, however, to also list on SYSOUT the records read in from the system input medium SYSDTA (data records, command and input character strings). This can be done by setting job switch 4 to on.

Each input record can be identified in the listing by the line preceding it, which contains a vertical line (|) in column 1.

Switch 5 set to ON for EDOR: unbuffered screen display

As a standard procedure, output resulting from the \$-command is buffered (page by page). In some cases, however, the user may want to have the individual output lines displayed immediately, that is, without buffering, e.g. if certain records in a file are to be sought and listed on the screen. This is effected by setting job switch 5 to on.



Unbuffered writing of output records is carried out via the system function WROUT. The exact effect of WROUT may vary according to the type of device used, so that, for example, on 8151 Data Display Terminals an ETX character is written after every record that is output.

Switch 6 set to ON for EDOR: alternative input for variable &

Input for the variable & is normally expected via the data display terminal, even if SYSDTA is reassigned and switch 1 is set to ON. Input for the variable & is switched to SYSDTA by setting switch 6 to ON (in addition to switch 1).

Switch 7 set to ON for EDOR: unsolicited programmed I/O

It is standard for texts in programmed input/output to be treated as "logical records". In order to separate them, EDOR expects or writes a (terminal-dependent) LZE. Additionally, on output, an overflow check is performed, providing the user with the option either to cancel the command and enter another one or to continue (message: "TYPE '+' TO CONTINUE OR NEW COMMAND").

For special applications, it is necessary that you program the processing and formatting of the entire screen contents to a large extent independently, i.e. without the insertion, removal, or interpretation of LZE characters and without overflow checking by EDOR. The latter requires unbuffered output.

The I/O mode described above is enabled by setting switches 5 and 7. Every output leaves the cursor located behind the last written character, without additional positioning. Any control characters X'OD' that may occur in the output text will be shown as LZE.

Switch 8 set to ON for EDOR: suppress start/end messages

In interactive processing, display of start and end messages is the default value in line mode. Setting switch 8 to on before EDOR is called will suppress the display of the start and end messages.

## **EDT**

Switch 4 set to ON for EDT: EDT suppresses first and last messages

Interactive mode, batch mode:

Setting job switch 4 before loading EDT suppresses load message BLS0500 and, after completion of EDT, message EDT800. The message "EDITED FILES NOT SAVED-TERMINATE (Y/N)" is also suppressed.

Batch mode:

Setting job switch 4 causes the operand NONE to be set in the @LOG statement, i.e. logging does not take place during the EDT run.

Switch 5 set for EDT: EDT operates in L mode

Input is read line by line from SYSDTA. When output to the screen, the current line number is replaced by '\*'. The @EDIT FULL SCREEN statement switches over to F mode. (Activating/deactivating switch 5 during the EDT run has no effect on the mode set.)

Switch 6 set for EDT: EDT outputs 160 characters

If this switch is set, EDT writes 160 characters in a SYSLST record (line) and any overflow to the next record. Use of this option is recommended if the SYSLST (system) file is to be output to a printer with a maximum line length of 160 characters. The switch must be set before calling EDT.

EDT usually outputs only 132 print positions and writes any overflow to the next record.

Switch 7 set for EDT: superfluous memory space is not released.

If switch 7 is set, the superfluous memory space assigned is not automatically released by EDT. EDT usually releases unoccupied memory space using a FILE command with a negative value for the SPACE operand. This switch can also be set while EDT is executing.

### **FMS**

Switch 5 set to ON for FMS: screen queries are suppressed by FMS in interactive mode.

### **LMS**

Switch 1 set to ON for LMS: alternative entry of instructions.

In interactive mode, it is standard for instructions to LMS to be read in from interactive terminals by means of the WRTRD macro. When switch 1 is set to ON, the instructions are read in by the RDATA macro from the file assigned to logical system file SYSDTA.

Switch 4 set to ON for LMS: suppress start/end message.

Setting switch 4 to ON suppresses the LMS start and end messages and keeps the LMS tracer listing to a minimum.

Switch 8 set to ON for LMS: suppress messages of the access routines.

Setting switch 8 to ON has the effect that the messages of the access routines will not be displayed.

Switch 9 set to ON for LMS: request additional space.

If switch 9 is set to ON, up to 12000 unequal records can be processed in one compare operation and larger directories can be sorted contiguously by the TOC function.

### **PRSERVE**

Switch 0 set to ON for PRSERVE: suppress output of character frames.

Setting switch 0 to ON suppresses the output of the secondary statement display (character frame) to SYSOUT. This does not, however, apply to screen output triggered by a DISPLAY statement.

Switch 2 set to ON for PRSERVE: suppress output of LOOP record.

Setting switch 2 to ON suppresses the output of the LOOP record (following the CHECK statement) in interactive mode.

Switch 3 set to ON for PRSERVE: suppress information on current operation mode.

It is standard for you to receive an error message and information on the current PRSERVE operating mode after entering an invalid PRSERVE statement. Setting switch 3 to on suppresses this information.

**STEP command**

Switches 16 to 31 are set to OFF when you issue a STEP command.

**SYSGEN/UGEN**

Switch 1 set to ON for SYSGEN/UGEN:

ENTER file SYSGEN sets switch 1 to ON before calling UGEN and checks its position when UGEN is finished.

The UGEN procedure sets switch 1 to OFF if it has been executed successfully. If switch 1 is still set to on, ENTER file SYSGEN reports the unsuccessful UGEN execution by a console message and terminates itself.

**TSOSLNK**

Switch 4 set to on for TSOSLNK: when printing with SPACE=E, control characters for page feeds are inserted according to the standard page format.

Switch 4 not set for TSOSLNK: when printing with SPACE=E, control characters for page feeds are inserted according to the logical structure of the linkage editor listing.

**MSGEDIT/MSGLIB**

Switch 1 set to ON for MSGEDIT/MSGLIB:

Setting job switch 1 to ON makes it possible to conduct a dialog with the runtime system of PLI1.

## SETUS Reset user switches

Application group: Job control (page 22 ff.)

### Command description

The SETUS enables you to set, reset or invert user switches.

32 user switches, numbered 0 through 31, are provided for every user ID. These switches are stored in the JOIN file of the home pubset.

Immediately after creation of a user ID, all its 32 switches are set to OFF. They subsequently retain the settings assigned by the user (even after LOGOFF!).

The user switches enable you to control job execution (SKIPUS and WHEN commands).

### Format and operand description

Operation	Operands
SETUS	$[\text{ON}=\left\{\begin{array}{l} \text{no} \\ (\text{no}, \dots) \end{array}\right\}]$ $[, \text{OFF}=\left\{\begin{array}{l} \text{no} \\ (\text{no}, \dots) \end{array}\right\}]$ $[, \text{INVERT}=\left\{\begin{array}{l} \text{no} \\ (\text{no}, \dots) \end{array}\right\}]$

**no**                      Number of a user switch to be altered. There are 32 user switches, numbered from 0 to 31.

**ON=no|(no,...)**      The specified switches are set to ON.

**OFF=no|(no,...)**     The specified switches are set to OFF.

**INVERT=no|(no,...)**  
 The specified switches are set to OFF if they were ON, and vice versa.



*Example:*

```

(IN)   GETUS _____ (01)
(OUT)   SWITCHES ON EQUAL- _____ (02)
(OUT)   1, 2, 3, 20
(IN)   SETUS ON=(5,7,18) _____ (03)
(IN)   GETUS _____ (01)
(OUT)   SWITCHES ON EQUAL-
(OUT)   1, 2, 3, 5, 7, 18, 20
(IN)   SETUS INVERT=(1,10,18) _____ (04)
(IN)   GETUS _____ (01)
(OUT)   SWITCHES ON EQUAL-
(OUT)   2, 3, 5, 7, 10, 20

```

- (01) The numbers of all user switches set to ON are to be output.
- (02) User switches 1, 2, 3 and 20 are set to ON.
- (03) Switches 5, 7 and 18 are set to ON, and then queried with the GETUS command.
- (04) Switches 1 and 18 are set to OFF because they were previously on; switch 10 is set to ON because it was previously set to OFF.

## SET-SS-OPTIONS      Release address space reserved for subsystems

Application group: Device and volume reservation (page 29)

### Command description

The SET-SS-OPTIONS command enables you to release space, for the duration of a task, by doing without a subsystem group for which the system administrator has reserved part of the class 5 memory with SCOPE=GLOBAL. For a more detailed description of DSSM, see the "System Installation" manual [13].

Entering the command cancels the reservation of the associated user address space and enables the address area to be used for other purposes (e.g. for a large application program).

### Format and operand description

Operation	Operands
SET-SS-OPTIONS	SS-NAME= { *NON-PRIV-SS *ADDR-REGION } , USAGE=NO

**SS-NAME**            Specifies the subsystems to be released.

  =\*NON-PRIV-SS All non-privileged subsystems are released.

  =\*ADDR-REGION

                  All privileged and non-privileged subsystems are released.

**USAGE=NO**            Subsystems will not be used. The user address space allocated for them may be used for other purposes.

## SHOW-CJC-STATUS      Display information on CJC functions

Application group: job variable functions (page 36)

### Command description

The SHOW-CJC-STATUS command provides information on jobs with conditional job control (CJC) which are currently waiting for events (changes in job variable values). CJC stands for Conditional Job Control.

The query can be restricted to the user's own processor but also be extended to any or all processors in an existing multiprocessor system. It is also possible to restrict the information to one or more specifically named job variables. In this case, only those jobs are displayed which have functions in which the specified job variables occur in conditional expressions.

Output is directed to SYSOUT.

For queries to foreign processors of an MSCF system, only those jobs are listed whose CJC functions use at least one job variable from a local catalog of the calling process in a conditional expression.

No JV link names may be specified at any point.

For further information on MSCF multiprocessor systems, see the "Multiprocessor Systems" manual [15].

Format and operand description

Operation	Operands
<pre> [ SHOW   -CJC   -STATUS ] [ S-C-S ] </pre>	<pre> [ HOST = {   *OWN   *ALL   *FOREIGN   'host'   ('host1', 'host2', ...)   jvidh   (jvidh1, jvidh2, ...) } ] [ *CATALOG ( CATALOG-ID = {   'catid'   ('catid1', 'catid2', ...)   jvidc   (jvidc1, jvidc2, ...) } ) ] [ , JV-IDENTIFICATION = {   *ALL   jvidi   (jvidi1, jvidi2, ...) } ] [ , INFORMATION = {   *SUMMARY   *USER-LIST } ] </pre>

- HOST** Specifies the processor(s) to which the STATUS query refers.
- =\*OWN** Default value: information on jobs with existing CJC functions at the user's own processor is displayed.
  - =\*ALL** Information on jobs with existing CJC functions on all processors in an MSCF multiprocessor system is displayed. If there is no integrated computer network, this operand has the same effect as the \*OWN operand.
  - =\*FOREIGN** Information on CJC functions on all processors in a multiprocessor system *except the user's own* is displayed.
  - = 'host'**
  - =( 'host1', 'host2', ... )** Specifies one or more BCAM processor names from an existing multiprocessor system. This restricts the query to jobs with CJC functions on the specified processors.

=jvidh  
=(jvidh1,jvidh2,...)

Specifies one or more fully-qualified names of permanent or temporary job variables containing the BCAM name of a processor from the MSCF system. This restricts the query to jobs with CJC functions on the specified processors.

```
*CATALOG (CATALOG-ID=
  {
    'catid'
    ('catid1','catid2',...)
  }
  jvidc
  (jvidc1,jvidc2,...)
)
```

The display is restricted to jobs with CJC functions on the processors with which the specified catalogs are associated.

= 'catid'  
=('catid1','catid2',...)

Directly specifies the catalog ID(s).

=jvidc  
=(jvidc1,jvidc2,...)

Fully-qualified name(s) of job variables containing solely a valid catalog ID in the first position of their value area.

## INFORMATION

=\*SUMMARY

Default value: for each processor, the number of jobs, the number of users and the referenced catalogs are displayed.

=\*USER-LIST

For each processor, the jobs are listed individually with TSN, user ID (under TSOS only) and referenced catalogs.

## JV-IDENTIFICATION

=\*ALL

Default value: information on all jobs with CJC functions is displayed.

=jvidi  
=(jvidi1,jvidi2,...)

The display is restricted to jobs which have CJC functions and use one of the specified job variables in conditional expressions. jvidi may be:

- a fully-qualified job variable name
- a partially qualified job variable name
- a :catid:

For examples refer to the "Job Variables" manual [11].

# SHOW-DEVICE-CONFIGURATION Request configuration information

Application group: Interrogation of current values (page 31)

## Command description

The SHOW-DEVICE-CONFIGURATION command is used to request information on the configuration and the availability status of hardware units.

## Format and operand description

Operation	Operands
<pre> { SHOW-DEVICE   -CONFIGURATION } { SH-DEV-CONF }         </pre>	<pre> UNIT={ mn       (mn1, ...) } [, INFORMATION={ STD                                    INNER                                    OUTER                                    ALL                                    PATH } ] CLASS={ ALL         unit-class } [, ATTRIBUTE={ ALL                                    ATTACHED                                    DETACHED                                    DETACH-PENDING } ] ATTRIBUTE={ ALL             ATTACHED             DETACHED             DETACH-PENDING }         </pre>

- ATTRIBUTE =ALL Standard information is displayed for hardware units in any state (default when only CLASS is specified).
- =ATTACHED Standard information is displayed for hardware units in the "attached" state.
- =DETACHED Standard information is displayed for hardware units in the "detached" state.
- =DETACH-PENDING Standard information is displayed for hardware units in the "detached-pending" state.

CLASS	Standard information is displayed for the hardware units of all device classes.										
=ALL											
=unit-class	Standard information is displayed for all hardware units associated with the specified device class. "unit-class" may be any of the following:										
	<table> <tr> <td>DVC</td> <td>Devices</td> </tr> <tr> <td>CTL</td> <td>Controls</td> </tr> <tr> <td>CHN</td> <td>Channels</td> </tr> <tr> <td>IOP</td> <td>I/O processors</td> </tr> <tr> <td>CPU</td> <td>Central processing units</td> </tr> </table>	DVC	Devices	CTL	Controls	CHN	Channels	IOP	I/O processors	CPU	Central processing units
DVC	Devices										
CTL	Controls										
CHN	Channels										
IOP	I/O processors										
CPU	Central processing units										
INFORMATION	Type of information desired. If CLASS and/or ATTRIBUTE are specified, only standard information is displayed.										
=STD	Standard information										
=INNER	In addition to the standard information, information on all internal connections of unit "mn" is displayed.										
=OUTER	In addition to the standard information, information on all external connections of unit "mn" is displayed.										
=ALL	In addition to the standard information, information on all internal and external connections of unit "mn" is displayed.										
=PATH	Information on possible I/O paths of terminal "mn" is displayed, together with their status.  INFORMATION=PATH returns information on terminals only.										
	<ul style="list-style-type: none"> <li>– No defined connections exist for device class CPU.</li> <li>– No internal connections exist for device class IOP.</li> <li>– No external connections exist for device class DVC.</li> </ul>										
UNIT=mn	Mnemonic device name (2 or 4 characters). Information is output on the specified devices. Up to 26 names may be specified.  Mnemonic device names that cannot be generated (beyond the admissible range of values) are treated like ungenerated mnemonic device names.										

**Alphabetical list of output fields**

Output field	Meaning
CTL-CHN-IOP	Virtual connection of device with IOP
CONF-STATE	Configuration status of specified device
DEV-TYPE	Name of device type
DVC	Mnemonic device name
ICUU	Path address (IOP, CHN, CTL, DVC) to one device
INNER CONNECTIONS	Status of connections with inner units
I/O-PATH-STATE	Availability of complete input/output path
MNEM	Mnemonic device name
OUTER CONNECTIONS	Status of connections with outer units
POOL	Availability of a device relative to two or more systems
UN-CLASS	Device class for specified unit
UN-TYPE	Device types and CTL, CHN, IOP and CPU types

The letter "S" appears between the UN-CLASS and UN-TYPE fields in the case of magnetic tape cartridge devices with stacker.

The output fields are described in detail in the appendix.



# SHOW-DEVICE-STATUS

## Request information on device assignment and monitoring

Application group: Interrogation of current values (page 31)

### Command description

The SHOW-DEVICE-STATUS command provides information on the specified devices and volumes. Information is displayed only with respect to those volumes which are physically online (as opposed to SHOW-DISK-STATUS). If no volume is online on the device, the display shows which volume is to be mounted on the device. Information is displayed only with respect to those devices and volumes which are occupied by the user task. There are no restrictions in terms of summary output (INFORMATION=SUMMARY).

During a REMOUNT operation, a VSN can appear simultaneously in two output records: in the record of the device where the volume is physically online, and in the record of the device where the volume is to be mounted.

### Format and operand description

Operation	Operands
<pre>{ SHOW-DEVICE-STATUS } { SH-DEV }</pre>	<pre>[ UNIT={ mn         (mn, ...) }   [ TYPE={ ALL           device-family           device-type           volume-type }     [ , ATTRIBUTE={ ALL                    attribute } ] ]   ATTRIBUTE={ ALL               attribute }   [ , INFORMATION={ STD                    SUMMARY                    ALL                    SHORT } ] ] ]</pre>

**ATTRIBUTE** Identifies the devices by means of device attributes.

**=ALL** Information is displayed on those devices which satisfy one of the following attributes.

**=attribute** Information is displayed on all devices with the specified attributes. "attributes" may be any of the following:

Attribute	Meaning
ATT[ACHED] DET[ACHED] DET[ACH]-P[ENDING]	} configuration state
FREE DMS EXCL[USIVE] PUB[LIC]	} device allocation state
SWITCH SHARE	} pool attribute

**INFORMATION** Specifies the type of information desired. If the operand UNIT=... is specified, only the operand values STD and SHORT are allowed for INFORMATION.

**=STD** Standard information is displayed.

**=SUMMARY** An overview of configuration and assignment states is output.

**=ALL** Display as for specification of STD and SUMMARY.

**=SHORT** Provides a short version of STD (without header line). When "device-family" and "device-type" are specified, an additional overview is displayed as in the specification of SUMMARY.

TYPE	Identifies the devices about which information is to be provided.
=ALL	Information about all devices is displayed.
=device-family	Information is displayed about all devices in the specified family. Permitted specifications for "device family" are contained in the device table in the appendix.
=device-type	Information is displayed about all devices of the specified device type. Permitted specifications for "device type" can be found in the "Device type" column of the device table in the appendix.
=volume-type	Information is displayed about all magnetic tapes of the specified type. Permitted specifications for "volume type" can be found in the table in the appendix.
UNIT	Specifies one or more devices via their mnemonic device names. If UNIT is specified, only operand values STD and SHORT are permitted for the operand INFORMATION.  Mnemonic device names that cannot be generated (beyond the admissible range of values) are treated like ungenerated mnemonic device names.
=mn	Mnemonic device name.
=(mn,...)	Up to 26 mnemonic device names can be specified in a list.

The output fields are described in further detail in the appendix.

### Examples

```
/SH-DEV-ST TYPE=PRINTER, INF=SUMMARY
```

```
/SH-DEV-ST TYPE=DISK, INF=SUMMARY
```

# **SHOW-DISK-DEFAULTS**

## **Interrogate default values for disk parameters**

Application group: Interrogation of current values (page 31)

### **Command description**

The SHOW-DISK-DEFAULTS allows the current values for disk allocation and control to be interrogated.

### **Format and operand description**

Operation	Operands
{ SHOW-DISK-DEFAULTS SH-DISK-DEF }	

The output fields are described in the appendix.

# SHOW-DISK-STATUS      Display allocation and disk parameters

Application group: Interrogation of current values (page 31)

## Command description

This command provides information about the allocation and the disk parameters for the specified disks, together with information about volume monitoring. The information displayed refers to the volume reserving the device, regardless of which volume is physically online (in contrast to SHOW-DEVICE-STATUS). The command only provides information about those devices reserved by the user task.

## Format and operand description

Operation	Operands
<pre>{ SHOW-DISK-STATUS } { SH-DISK }</pre>	<pre>UNIT={ mn       { (mn, ...) } } [ VOLUME={ vsn           { (vsn, ...) } } ATTRIBUTE={ ALL             { attribute } } [ , INFORMATION={ STD                  { PARAMETER } } ]</pre>

**ATTRIBUTE**      Restricts the information displayed to a subset of all disks, i.e. those with the specified attribute. **ATTRIBUTE=...** must be specified in conjunction with **INFORMATION=STD**. This causes a record containing global allocation and monitoring information to be output for each disk of the subset.

**=ALL**            A record is output for each disk with at least one of the attributes listed below. Default value.

=attribute      A record is output for each disk with the specified attribute.

attribute	Meaning
FREE EXCL[USIVE] SHARE	} volume allocation state
PUB[LIC]	} device allocation state
ONLINE MOUNT IN-USE	} volume phase
CAN[CELED] NO-DEV[ICE] REC[OVER] DISMOUNT UNLOCK SVL-UPD[ATE]	} action state
DMS SPECIAL	} use mode
STD N[ON]-STD BS1000	} label type

**INFORMATION**      Specifies the type of information desired.  
 This operand is only valid in conjunction with UNIT or VOLUME. If  
 ATTRIBUTE is entered, only INFORMATION=STD is possible.

=STD      A record with global allocation and monitoring information is  
 displayed for each disk.

=PARAMETER      The parameters preset with the SET-DISK-PARAMETER or SET-DISK-  
 DEFAULTS command are displayed. Parameter values preset with  
 SET-DISK-DEFAULT are identified by a "(D)". For public disks, no  
 information can be requested with these operands.

If INFORMATION=PARAMETER is specified for PUBLIC disks, only  
 the mnemonic device name "mn" and the volume serial number  
 (VSN) are displayed.

If a number id disks with the same VSN exist in one system, only  
 one record is displayed if INFORMATION=PARAMETER is specified.

UNIT	Specifies one or more disk units via their mnemonic device names. Mnemonic device names that cannot be generated (outside the permissible range of values) are treated like ungenerated mnemonic device names.
=mn	Mnemonic device name of the disk device.
=(mn,mn,...)	List of up to 26 mnemonic device names.
VOLUME	Indicates the disks for which information is to be displayed. Wildcards may be substituted for any part at the beginning or end of the volume serial number to be specified. In this case, information is displayed about all disks whose VSN matches the specified partial VSN. The asterisk ("*") should be used as the wildcard. Specification of the asterisk all by itself causes information about all disks to be displayed that are either monitored by NDM or whose parameter values have been set by means of the SET-DISK-PARAMETER command.
=vsn	Volume serial number of the disk.
=(vsn,vsn,..)	List of up to 10 volume serial numbers.

The output fields are described in further detail in the appendix.

## SHOW-FILE      Display file or library member on screen

Application group: File processing (page 26 ff.)

### Command description

The SHOW-FILE command allows you to display a file (SAM, ISAM and PAM files) or a member of a PLAM library on the screen without having to load an editor (editing program).

Once the command is entered, the specified file or library member is opened, and the first section is output to the terminal. You may then enter further instructions, e.g. to scroll the file or to search for a string.

The command is effective in interactive mode only.

### Format and operand description

Operation	Operands
$\left\{ \begin{array}{l} \text{SHOW-FILE} \\ \text{SH-F} \\ \text{S-F} \end{array} \right\}$	$\text{FILE-NAME} = \left\{ \begin{array}{l} \text{pathname1} \\ * \text{LIBRARY-ELEMENT} (\text{LIBRARY} = \text{pathname2}, \text{member}) \end{array} \right\}$ $[ , \text{OUTPUT-FORMAT} = \left\{ \begin{array}{l} \text{STD} \\ \text{DUMP} \\ \text{CHARACTER} \\ \text{HEX} \end{array} \right\} ]$

pathname1      stands for: [:catid:][userid.]filename

catid            Catalog ID of the pubset on which the file is stored.  
Default value: the catalog ID assigned to the user ID.

userid            User ID to which the file is assigned.  
Default value: the user ID from the LOGON command.

filename        Fully-qualified file name of the file to be output.

\*LIBRARY-ELEMENT(...)  
Denotes the member to be output from a PLAM library (see the "LMS" manual [14]).

pathname2      stands for: [:catid:][userid.]library  
See above for "catid" and "userid".



library (fully-qualified) name of the (PLAM) library containing the member to be output.

member stands for:

$$\text{ELEMENT}=\text{membername} [(\text{VERSION}=\left\{ \begin{array}{l} *STD \\ \text{version} \end{array} \right\} [, \text{Variant}=\left\{ \begin{array}{l} STD \\ \text{var} \end{array} \right\} ] ) ], \text{TYPE}=\text{type}$$

membername Name of the library member (up to 64 characters).

version Version number of the member (up to 24 characters).

\*STD Default value: highest version number.

VARIANT Variant number of the member (max. 4 digits)  
=var

STD The variant with the highest number of the member that is specified by type, name and version; default value.

type Denotes the type of member. Load modules (C-type members) cannot be output, and records of a member that exceed 2 Kbytes in length ("B" records) are not output in their entirety.

#### OUTPUT-FORMAT

Specifies the output format of data (character constants [printable characters], hexadecimal constants, dump format).

=STD Default output format: SAM files, ISAM files, and PLAM library members are output as character constants; PAM files are output in dump format.

=DUMP Dump format:  
The file to be output is opened using the PAM access method. It is output in units of 2 Kbytes (PAM pages). Each line begins with a 4-digit hexadecimal number, the byte number of the first data byte on the line within the current PAM page. The byte number is followed by data bytes in the form of hexadecimal constants, and then displayed as printable characters. This entry is not permitted for PLAM library members.

=CHARACTER

Data is displayed as character constants; non-printable characters are represented as smudge characters or as characters set with /TCHNG SUB. The display can be activated using the HEX ON statement.

=HEX

Data is displayed on 3 lines. 1st line: Data is displayed as  
constants. 2nd line: Hexadecimal encoding of the 1st half byte of  
the character constants on line 1.  
3rd line: Hexadecimal encoding of the 2nd half byte of the  
character constants on line 1.

The display can be deactivated using the HEX OFF statement.

### Screen display

Contents of file or PLAM library member	
Statements	Status indicator

The contents of the file or PLAM library member are displayed in the first 23 of the 24 screen lines. The last line of the screen contains no data; this line is the statement and status line. The left-hand part of the line may be used to enter statements; the right-hand part of the line contains the status indicator.

## Status indicator

The status indicator provides information on the file type and positional attributes of the file section displayed. The display is in the form:

```
type * reference direction recordno recordpos.
```

where:

type	Specifies the file type (I=ISAM, S=SAM, P=PAM, L=PLAM).
reference	Reference position (SOF=start of file, EOF=end of file, KEY=last record searched with ISAM key). For PLAM files, the record type is shown here in hexadecimal notation.
direction	Direction in which paging may take place (+ or -). This entry is irrelevant for PLAM files (represented by "/").
recordno	Record number relative to the reference ("reference") position. For PAM files the PAM page number is displayed; for PLAM files the record number within the indicated record type.
recordpos	Record position specifies the column number of the first character on the screen. For PAM files, the position within the PAM page is displayed.

## Example

```
S*SOF+      1 (  1)
```

## Entering statements

One or more statements can be written in the statement line. If more than one statement is entered, they must be separated by semicolons. When the statement line is not adequate, for example, while searching for a string, a continuation line is provided by pressing the DÜ key (data transmission key)

Statements can also be input via a file assigned to SYSDTA, provided that switch 5 has been set to on. The maximum record length is 80 characters. Statements must begin in the second column. If the last statement of a record is a character string, the statement must end with a semicolon. The following record is interpreted as a continuation line, if the previous statement was incomplete (e.g. missing apostrophe, missing parenthesis).

If an error occurs while processing a statement, an error message appears in the second-last line of the screen, and processing is aborted. The statement containing the error is output in the statement line. If the invalid statement is part of a sequence of statements, the unprocessed part of the sequence is also displayed in the statement line, in addition to the invalid statement.

The HELP command cannot be used (for an explanation of the error message), as long as the file is still being displayed.

The K3-key can be used to restore screen contents if, for example, the screen has been destroyed by a BCST-message from the operator.

#### *Vertical scrolling*

- ++            Position to end-of-file
- Position to begin-of-file
- +*[n]*        Move forward by "n" records; default value:n=screen height.  
With OUTPUT-FORMAT=DUMP: move forward by "n" PAM pages;  
default value: n = 1. The first 368 bytes (=170<sub>16</sub>) of the PAM page are  
output. Movement within the PAM page is accomplished with >*[n]*,  
<*[n]* or <<.
- [n]*        Move backward by "n" records; default value: n=screen height.  
With OUTPUT-FORMAT=DUMP: move backward by "n" PAM pages;  
default value: n = 1.

#### *Horizontal scrolling*

- <<            Move left, to the beginning of the record.
- <*[n]*        Move left by "n" columns; default value: n=screen width.  
With OUTPUT-FORMAT=DUMP: move forward by n bytes in a PAM  
page; default value: n = 368 (=170<sub>16</sub>).
- >*[n]*        Move right by "n" columns; default value: n=screen width.  
Move backward by "n" bytes in a PAM page; default value: n = 368  
(170<sub>16</sub>).
- S*[n]*        Positioning to the nth column, or - for dumps - to the nth byte, within a  
PAM page; default value: n=1.
- = 1.

*Positioning to a specific block/record of the file*

- Rn                    Positioning to the nth PAM page of a file opened with PAM or to the nth record of a file not opened with PAM.
- K[C]'isamkey'        "isamkey" is the record key; specified as a character constant. Data output begins at the specified record. Apostrophes within the string must be doubled; lowercase letters are interpreted as uppercase.
- KX'isamkey'         "isamkey" is the record key; specified as a hexadecimal constant. Data output begins at the specified record.

*Display (new) library member*

([type[,membername[,version]])

See above for "type", "membername" and "version". The specified member of the open PLAM library is displayed. Any member which was opened previously is automatically closed. Default values:

- type                    = S-type member (source program).
- member name         = last member of the specified type entered into the library.
- version                = highest version for the specified member.

If the user fails to enter any parameter, the default value is assumed, even for parameters to the right

*Search character string*

- [C]'string'            String of characters for which the file is searched; specified as a character constant. "string" must not be longer than 256 bytes. Further output of data starts at the hit record.
- X'string'             String of characters for which the file is searched; specified as a hexadecimal constant. Further output of data starts at the hit record.



If the string is found, the current position is set to the beginning of the hit record; in the case of a dump format, to the first byte of the string.

If a hit is not encountered while searching for the string, a warning is displayed. The current position remains unchanged in this case.

*Modify output format*

- HEX ON        The output is the same as for OUTPUT-FORMAT=HEX.  
HEX OFF       The output appears in the original format.

*Terminate file output*

- END            The file or library member is closed. You can enter system commands once more.

**Interblock gaps**

Files with BLKCTRL=PAMKEY and BLKCTRL=DATA can possess so-called interblock gaps, i.e. logical blocks which are in fact reserved for a file but are not yet currently occupied. These logical blocks are recognized by means of their invalid CFIDs. As these blocks may still contain data that does not belong to the file, the PAM pages of such blocks will be output as "empty" PAM pages (2048 \* X'00'). In addition, a message is issued pointing out that

- the currently displayed PAM page (OUTPUT-FORMAT=DUMP) or
- one or more PAM pages (in the case of PAM files and OUTPUT-FORMAT=CHAR or HEX)

is/are not occupied.

This applies to all PAM files (regardless of the OUTPUT-FORMAT) and to all ISAM and SAM files with OUTPUT-FORMAT=DUMP.

## SHOW-FILE-TRANSFER      Display information about FT jobs

Application group: File transfer (FT) (page 37)

This command is only available with the FT software product.

### Command description

The SHOW-FILE-TRANSFER command supplies information on all FT jobs being processed under the user's own ID. The following items of information are displayed for each job:

- FT job number (transfer ID)
- command initiator
- processing state (ACTIVE, FINISHED, WAIT, HOLD, ...)
- partner system
- direction of transfer
- number of bytes transferred
- file or member name in the local system
- specifications made in the TRANSFER-FILE command

The query can be restricted to a subset of FT jobs by specifying selection criteria.

This section contains only an outline description of the SHOW-FILE-TRANSFER command. For a complete description and explanation of all mnemonic operands (long form) please refer to the "File Transfer" manual [7], which also provides examples of applications. (Valid for Version FT-BS2000 V4.0B).

Format (short form) and operand description

Operation	Operands
$\left\{ \begin{array}{l} \text{SHOW-FILE-} \\ \text{TRANSFER} \\ \text{NSTATUS} \end{array} \right\}$	$[\text{TRANS}=\left\{ \begin{array}{l} *ALL \\ \text{no} \end{array} \right\}]$ $[\text{, SELECT}=\left\{ \begin{array}{l} *OWN \\ (\text{INIT}=\left\{ \begin{array}{l} (*LOCAL, *REMOTE) \\ *LOCAL \\ *REMOTE \end{array} \right\}) [\text{, PARTNER}=\left\{ \begin{array}{l} *ALL \\ \text{name} \end{array} \right\}] \end{array} \right\}]$ $\rightarrow [\text{, FILE-NAME}=\left\{ \begin{array}{l} *ALL \\ *LIB (\text{LIB}=\left\{ \begin{array}{l} *ALL \\ \text{library} \end{array} \right\}) \\ \text{filename} \end{array} \right\}, \text{ELEM}=\left\{ \begin{array}{l} *ALL \\ \text{member} \end{array} \right\} (\text{VERS}=\left\{ \begin{array}{l} *ALL \\ \text{version} \end{array} \right\}) \}, \text{TYPE}=\left\{ \begin{array}{l} *ALL \\ \text{type} \end{array} \right\}]$ $\rightarrow [\text{, MONJV}=\left\{ \begin{array}{l} *NONE \\ \text{jvname} \end{array} \right\}], [\text{JV-PASS}=\left\{ \begin{array}{l} *NONE \\ \text{password} \\ *SECRET \end{array} \right\}] [\text{, STATE}=\left\{ \begin{array}{l} *WAIT \\ *ACTIVE \\ *FINISHED \\ *HOLD \end{array} \right\} \}]]$ $[\text{, INF}=\left\{ \begin{array}{l} *STD \\ *ALL \end{array} \right\}] [\text{, OUTPUT}=\left\{ \begin{array}{l} *SYSOUT \\ *SYSLST \end{array} \right\}]$

INF Determines the amount of information output.

- =STD The following items of information are output per job:
  - FT job number (transfer ID)
  - command initiator
  - processing state (SUSPEND, LOCK, HOLD, ...)
  - partner system
  - direction of transfer
  - number of bytes transferred
  - file or member name in the local system

=\*ALL Specifications made in the TRANSFER-FILE command are output in addition to the information output with \*STD.

OUTPUT Determines whether information is output to the SYSOUT system file or to SYSLST.

=SYSOUT Output will be to SYSOUT; default value.

=\*SYSLST Output will be to SYSLST.



SELECT={...}	Selects a subset of FT jobs. For a job to be included, all selection criteria specified must apply to it.
<u>*OWN</u>	Information about all your FT jobs is to be displayed; default value.
INIT	Selects those FT jobs which have been started in the local or the remote system. The local system is the one where the <b>NSTATUS</b> command was issued.
<u>=(*LOCAL,*REMOTE)</u>	Jobs are selected without taking the initiating system into account; default value.
=*LOCAL	All jobs started in the local system are selected.
=*REMOTE	All jobs started in the remote system are selected.
PARTNER	Selects those FT jobs processed in cooperation with the specified partner system.
<u>=*ALL</u>	Jobs are selected without taking the partner system into account; default value.
=name	Name of the partner system.
FILE-NAME	Selects those FT jobs which access the specified file/ library. The file/library must be cataloged in the local system.
<u>=*ALL</u>	Jobs are selected without taking the name of the file/ library into account; default value.
<u>=*LIB(...)</u>	Selects those FT jobs which access the specified library. Selection of the library can be further restricted by specifying a library name/member ("ELEMENT") name/ member version/member type.
<u>*ALL</u>	Jobs are selected without further restriction via a library name/member name/member version/member type.
library	library name
member	member name
version	version ID
type	member type
=filename	File name.
MONJV	Selects those FT jobs that are monitored by a job variable.
<u>=*NONE</u>	Jobs are selected without taking a monitoring job variable into

	account.
=jvname	Job variable name.
JV-PASS	Specifies the password protecting the job variable.
=* <u>NONE</u>	No password has been defined.
=password	Password protecting the job variable.
=*SECRET	The system requests your password. Screen display of the password upon input is suppressed.
STATE	Selects those FT jobs that are in a specified processing state.
=* <u>ALL</u>	Jobs are selected without taking the processing state into account; default value.
=*SUSPEND	All jobs in the "suspend" state (=interrupted) are selected.
=*LOCKED	All jobs in the "lock" state (=temporarily locked due to a resource bottleneck) are selected.
=*WAIT	All jobs in the "wait" state (=waiting for resources) are selected.
=*ACTIVE	All jobs in the "active" state (=being processed) are selected.
=*FINISHED	All jobs in the "finished" state (=terminated or aborted without notice) are selected.
=*HOLD	All jobs in the "hold" state are selected.
TRANS	Specifies the FT job about which information is requested.
=* <u>ALL</u>	Information about all your FT jobs is to be displayed; default value.
=no	FT job number (transfer ID).

## SHOW-JOB-CLASS Request information on job classes in use

Application group: Interrogation of current values (page 31)

### Command description

The SHOW-JOB-CLASS command allows you to request description of all the job classes to which you have access.

If the name of a job stream is displayed for an interactive job class, this is insignificant because interactive jobs are not subject to scheduling.

### Format and operand description

Operation	Operands
$\left\{ \begin{array}{l} \text{SHOW-JOB-} \\ \text{CLASS} \\ \text{SH-J-C} \end{array} \right\}$	$[\text{NAME}=\left\{ \begin{array}{l} \text{*ALL} \\ \text{*ALL-NAMES} \\ \text{name} \\ \text{(name, . . .)} \\ \text{(name, . . .)} \end{array} \right\}] [\text{, OUTPUT}=\left\{ \begin{array}{l} \text{SYSOUT} \\ \text{SYSLST} \end{array} \right\}]$

<b>NAME</b>	Determines the extent of information to be displayed.
=* <u>ALL</u>	Displays information concerning all the job classes to which you have access (default value).
=*ALL-NAMES	Displays only the names of job classes to which you have access.
=name	Specifies the name of the job class on which information is desired.
=(name,...)	Specifies the names of two or more job classes on which information is desired.
<b>OUTPUT</b>	Defines where the information is to be output.
= <u>SYSOUT</u>	Output is sent to the data display terminal where the command was entered (default value).
=SYSLST	Output is to SYSLST. This entry is advisable when output is extensive.

## SHOW-JV-LINK      Display job variable entries

Application group: Job variable functions (page 36)

The SHOW-JV-LINK command is only available with the JV software product (see also the "Job Variables" manual [11]).

### Command description

The SHOW-JV-LINK command enables you to display the entries in the JV link table for your job. The assignments between job variable names and their link names are entered in the JV link table. These assignments are made by means of the DCLJV command.

### Format and operand description

Operation	Operands
$\left\{ \begin{array}{l} \text{SHOW-JV-LINK} \\ \text{SH-J-L} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{JV-NAME} = \left\{ \begin{array}{l} \text{*ALL} \\ \text{JVNAME} \end{array} \right\} \\ \text{LINK-NAME} = \text{*JVLINK} \end{array} \right\}$

<b>JV-NAME</b> =*ALL =JVNAME	All job variable entries for this job are sorted according to link name and output to SYSOUT.  All entries associated with the job variable "JVNAME" are output to SYSOUT. "JVNAME" is the fully or partially qualified name of a permanent or temporary job variable. The use of wildcards is permissible.
<b>LINK-NAME</b> =*JVLINK	Specifies the link name (previously assigned in a DCLJV command) whose entry is to be output.  A JV link name is always uniquely associated with a job variable name; conversely, a JV name may be connected with more than one link name.

For examples, see the "Job Variables" manual [11].

## SHOW-MOUNT-PARAMETER      Display MOUNT settings

Application group: Interrogation of current values (page 31)

### Command description

The SHOW-MOUNT-PARAMETER command provides information on the settings for mounting and dismounting tapes and disks.

### Format and operand description

Operation	Operands
{ SHOW-MOUNT -PARAMETER } [SH-MOUNT-PAR]	

### Alphabetical list of output fields

Output field	Value	Meaning
ALLOCATE-TAPE	YES/NO	Automatic assignment of volumes
DISK-MOUNT	YES/NO	MOUNT status of operator
TAPE-MOUNT	YES/NO	MOUNT status of operator
UNLOAD-RELEASED-TAPE	YES/NO	Automatic unloading of tapes

## SHOW-MSG-DEFAULTS      List message files

Application group: Job control (page 22 ff.)

### Command description

The SHOW-MSG-DEFAULTS command allows you to request information on message files. The following data is displayed:

- number of message files (system-wide, task-specific)
- language used for message output (system-wide, task-specific)
- names of message files; each name is followed by an indicator of the scope and access method:
  - S = (system-wide) message file
  - T = (task-specific) message file
  - D = DLAM (Direct Logical Access Method  $\triangleq$  optimized ISAM)
  - I = ISAM
  - D+I = DLAM + ISAM

System message files are listed first, followed by the task-specific message files.

Message files can be assigned a scope (system-wide or task-specific). You can use your own message files for message output restricted to your own tasks. In addition, it is possible to specify the language which is to be given preference when selecting the message texts for output. Message files and the language specification are included in the message system by means of the MSGCONTROL command.

### Format and operand description

Operation	Operands
SHOW-MSG-DEFAULTS	SCOPE={ BOTH SYSTEM TASK }

**SCOPE**                      Specifies whether system-wide or task-specific message files are to be listed.

**=BOTH**                      Both system-wide and task-specific message files are listed; default value.

**=SYSTEM**                    Only the system-wide message files are listed.

**=TASK**                      Only the task-specific message files are listed.

**Example**

```
(IN)  SHOW-MSG-DEFAULTS _____ (01)
(OUT) NUMBER OF SYSTEM MESSAGE FILES : 10 _____ (02)
      NUMBER OF TASK MESSAGE FILES : 2
```

```
SYSTEM LANGUAGES : ED
TASK LANGUAGE   :     TASK SEARCH : *TASK
```

MESSAGES-FILES NAMES	OPEN
:Q:\$B09.SYSMSA.GENSKP.010	S I
:D:\$TSOS.SYSMSA.TSADMIN.010	S I
:Q:\$RZ2.SYSMSA.UDS	S I
:Q:\$RZ30.SYSMSA.SESAM	S I
:Q:\$RZ20.SYSMSA.SESDCN	S I
:Q:\$RZ20.SYSMSA.SESAM	S I
:Q:\$RZV100.SYSMSG.A.DSSMGEN.2.0A41	S I
:D:\$TSOS.SYSMSG.A.OSS	S D+I
:D:\$TSOS.SYSMSA.SRPMCARD.010	S I
:D:\$TSOS.SYSMSA.SDF.014	S I
:M:\$USER.MESSAGES.100.03	T I
:M:\$USER.MESSAGESNP100.04	T I

- (01) The SHOW-MSG-DEFAULT command provides information on the number, names and language of the (system-wide) and (task-specific) message files.
- (02) 10 (system-wide) and 2 (task-specific) message files are listed. Both English (E) and German (D) were defined at system generation time as languages to be used for message output. Messages are output in German unless the message text exists in English. No language has been defined for task-specific message output. The remainder of the output is a list of the names of system-wide and task-specific message files; the right-hand columns contain the symbols for scope and access method.

# SHOW-RESOURCE-ALLOCATION

## Display allocations and outstanding operator actions

Application group: Interrogation of current values (page 31)

### Command description

The SHOW-RESOURCE-ALLOCATION command allows you to request information on resource allocation and outstanding operator actions for a task running under your user ID.

In the case of DRV-DUAL assignments, the mnemonic device name of the second disk is output instead of the device allocation type.

### Format and operand description

Operation	Operands
$\left\{ \begin{array}{l} \text{SHOW-RESOURCE} \\ \text{-ALLOCATION} \\ \text{SH-RES} \end{array} \right\}$	$\left[ \begin{array}{l} \left\{ \begin{array}{l} \text{TSN} = \left\{ \begin{array}{l} *OWN-TSN \\ \text{tsn} \end{array} \right\} \\ \text{MONJV} = \text{monjv} \end{array} \right\} \\ \\ \left[ \text{, IDENTIFICATION} = \left\{ \begin{array}{l} \text{JOB-NAME} \\ \text{USER-IDENTIFICATION} \end{array} \right\} \right] \\ \\ \left[ \text{, INFORMATION} = \left\{ \begin{array}{l} \text{RESOURCES} \\ \text{ACTIONS} \end{array} \right\} \right] \end{array} \right]$



## IDENTIFICATION

Controls allocation of the output field NAME/ID.

=JOB-NAME The job name is entered in the field NAME/ID (default value).

=USER-IDENTIFICATION

The user ID is entered in the field NAME/ID.

INFORMATION Defines the type of information desired.

=RESOURCES

A record is output regarding every resource allocation of the specified task (default value).

=ACTIONS

A record is output for each open operator action for a volume of the specified task. Operator actions include: mount volume, mount write-enable ring, cancel INOP, premount volume, remount volume, etc. The operating system only outputs this information if there is an operator action outstanding.

MONJV=monjv Identifies a task by means of its monitoring job variable.

TSN=\*OWN-TSN

Provides information for the user's own TSN (default value).

=tsn

Identifies a task by means of its TSN.

The output fields are described in further detail in the appendix.

**SHOW-SPOOL-CHARACTER-SETS****Interrogate character sets**

Application groups:

Interrogation of current values (page 31)

SPOOL jobs (page 35)

**Command description**

The SHOW-SPOOL-CHARACTER-SETS command provides information on the character sets defined for RSO printers. A list including character set name, font, weight, color, etc., is provided; see the list of output fields below. The description is valid for SPOOL V2.5B / RSO V2.1B .

**Format and operand description**

Operation	Operands
$\left\{ \begin{array}{l} \text{SHOW-SPOOL-CHARACTER-SETS} \\ \text{S-S-C} \end{array} \right\}$	$\text{CHARACTER-SET-NAME} = \left\{ \begin{array}{l} \text{*ALL} \\ \text{id} \\ \text{(id, ...)} \end{array} \right\}$

**CHARACTER-SET-NAME**

Identifies one or more character sets on which information is to be provided.

=\*ALL

Lists all existing character sets.

=id

Name of character set; "id" consists of 1-3 alphanumeric characters - with wildcards up to 24 alphanumeric characters.

=(id,...)

Up to 255 names can be specified in a list.

**Alphabetical list of output fields**

Output field	Meaning
C-S-NAME	Character set name.
CH-STYLE	Style of characters (STRAIGHT, ITALICS, ...).
CH-TYPE	Type of characters (OCR-A, OCR-B, ...).
COLOUR	Color.
CPI	Character density (characters per inch).
LANGUAGE	Language for the character set (ENGLISH, ...).
LPI	Line density (lines per inch).
NLQ	Near-letter-quality.
OWNER	Non-privileged owner: blank. Owner of character set: own user ID. TSOS: user ID of the device or system administrator responsible for generating the character set.
UND	Underscore.
WEIGHT	Type weight.

## SHOW-SPOOL-DEVICES      List printers

Application group: Interrogation of current values (page 31)

### Command description

The SHOW-SPOOL-DEVICES command can be used to request information on RSO and local printers, e.g. device name, device type, maximum number of characters in a print line, type of paper feed, position of character sets; see the list of output fields below. The description is valid for SPOOL V2.5B / RSO V2.1B .

### Format and operand description

Operation	Operands
{ SHOW-SPOOL-DEVICES } { S-S-D }	DEVICE-NAME= { <u>*ALL</u> id (id, ...)           }

- DEVICE-NAME**      Specifies one or more printers on which information is to be provided.
- =\*ALL**              Lists information about all assigned printers.
- =id**                    Name of the device (printer). "id" consists of 1-8 alphanumeric characters - with wildcards up to 24 characters.
- =(id,id,...)**        Up to 255 printers may be specified in a list.

## List of output fields (in alphabetical order)

Output field	Meaning
ACC	Printer connection (hardcopy, direct, STD, application).
BAND-ID	Name of the type band (9645 Printer) or *NONE.
BLANK-COMP	Indicates whether a string of blanks in printer memory can be abbreviated.
BUFFER-SIZE	Maximum size of the data blocks transferred to the printer.
CONTR-RES	Indicates whether a separate RSO controller is available for the printer (YES/NO).
CHAR-IMAGE	Names of the character code tables.
C-S-N	Number of the character set
DEF-TRAY	Number of the tray for single-sheet feeding.
DEV-NAME	Device name
DISCONN	Indicates whether the connection between RSO and the printer is interrupted if no RSO jobs are pending (YES/NO).
DUP-PROC	Specifies whether both sides of the page are to be printed
FO-F	Paper feed. SING = single sheet, LIST = continuous paper.
FORM-NAME	Name of the standard form.
F-O-B	Size of forms overlay
FORM-OVER	Indicates whether the printer allows forms overlay
IDENTIFICATION	Address of the device manager to be contacted (name, telephone number, etc.).
LI-FEED-COMP	Conversion of line feed control characters to printer control characters (YES/NO).
LI-S	Maximum number of characters per line.
LOAD	Indicates whether or not a (loadable) LOOP record must be specified for printing.
MODULO2	Specifies whether the contents of a second buffer are to be transferred while the contents of the first buffer are being printed
OUT-TRAY	Number of out-tray
POS-1	Position (POS-1, POS-2, ..., POS-16) of up to 16 character sets. Each character set (z.B. OCR-A, DATA,...) is located under the corresponding position number.
...	
POS-16	
PROC-NAM	Processor name (part of the device address or device management address)
REPEAT-MESS	Indicates whether a message to the printer will be repeated in the event of error (YES/NO).
RESET	Indicates whether the printer switch is to be reset before printing the file.
ROT	Specifies page rotation
R-P-M	Size of the raster pattern memory containing forms overlays and character sets
SHIFT	Default value for the SHIFT operand in the PRINT command.
SKIP-TO-CH	Skip-to-channel record
STAT-NAM	Name of the terminal (part of the device address or the device administrator address).
ST-FORM-FEED	Indicates whether page feed is performed automatically before starting the printout (YES/NO).
TIMOUT-MAX	Maximum printing time (in minutes) for a buffer.
TYPE	Printer type, e.g. 9001RP.
USER-IDENTI...	TSOS: device administrator ID or IDs. Non-privileged user: the output field will contain blanks.

## SHOW-SPOOL-FORMS List print forms

Application group: Interrogation of current values (page 31)

### Command description

This command provides information on all forms assigned to the various printer types (RSO and local printers). The list includes the name of the form, the character set and LOOP record assigned to the form, the assigned printer type, etc. (see the list of output fields below). The description is valid for SPOOL V2.5B / RSO V2.1B.

### Format and operand description

Operation	Operands
{ SHOW-SPOOL-FORMS } { S-S-F }	[ FORM-NAME = { <u>*ALL</u> id (id, ...) } ] [ , DEVICE-TYPE = { <u>ALL</u> type (type, ...) } ]

<b>FORM-NAME</b>	Name of the form on which information is to be provided.
= <u>*ALL</u>	Information is output on all forms assigned to the specified printer.
=id	Name of the form about which information is to be provided. "id" is an alphanumeric name with 1...24 characters including wildcards.
=(id,...)	Up to 255 forms can be specified in a list. "id" is alphanumeric with 1...6 characters.
<b>DEVICE-TYPE</b>	Designates the device type (printer type). Information is output on all forms assigned to this printer type.
= <u>ALL</u>	All connected printer types and the forms assigned to them are listed; default value.
=type	Printer type (1...16 characters).
=(type,...)	Up to 27 printer types can be specified in a list.

## Alphabetical list of output fields

Output field	Meaning
BAND-ID	Name of band used for printing
C-P	Position (line number) of channel 1 in the assigned LOOP record.
C-S	Character set assigned to the form by default.
DEV-TYPE	Name of the printer type.
FORM-NAM	Name of the form.
H-P	SPOOL header page (HEADER-PAGE). YES = 2 header pages, NO = no header page, ONE = 1 header page, RES = 1 header page after a restart.
LI-S	Line length in 1/10 inches.
L-N	Name of the LOOP record assigned to the form by default.
LPI	Number of lines per inch.
OWNER	TSOS: user ID of the device or system administrator who defined the form. Non-privileged user: the output field will contain blanks.
PA-DEF	Preset PCL file for the LP65 printer
PA-S	Page length in 1/10 inches
PREFORM	Control character string at the beginning of every print job
ROT-CONTROL	The fields list information on the page and line feeds and on the character set in the case of rotation (ROTATION).
S-O	Specifies whether print jobs are to be separated by an offset
S-P-T	Specifies whether print jobs are to be separated by a page feed.
T-P	SPOOL trailer page (TRAILER-PAGE). YES: Basic spoolout information and the PRINT command are printed on the trailer page. INFO: Only the basic spoolout information is printed on the trailer page. NO: The trailer page is not printed
T-UP-P	Specifies whether both sides are to be printed with the forms overlay
VERT-CONTROL	The fields list information on the page and line feeds and on the character set in the case of standard output (no rotation)

The L-N and LPI fields are mutually exclusive. One or the other is always empty.

## **SHOW-SPOOL-PARAMETERS      Show parameter list**

Application group: Interrogation of current values (page 31)

### **Command description**

The SHOW-SPOOL-PARAMETERS command allows you to request the current SPOOL parameter list. The description is valid for: SPOOL V2.5B / RSO V2.1B .

### **Format**

Operation	Operands
{ SHOW-SPOOL-PARAMETERS } S-S-P	



**Meaning of the output parameters:**

The parameters correspond to the operands of the MODIFY-SPOOL-PARAMETERS and SDVC commands (see the "SPOOL System" manual [19]). Several parameters define default values for operands of the PRINT/PUNCH commands and output values for the STATUS command.

Output field	Meaning
VERSION	Name of the SPOOL and RSO versions loaded.
HEADER-PAGE	
SPAC-AFT-HEAD	Indicates whether or not a branch is made to channel 1 after output of the header page.
FAM-MEMB-HEAD	Specifies whether a header page is to be printed for each file in a group of files.
COPY-HEADER	Specifies whether a header page is printed for each printed copy of a spoolout job.
CHARACTER-SET	Specifies whether the standard character set or the character set specified in the PRINT command is to be used to print the header page.
PRINT-ACC-NUM	Specifies whether the account number is to be printed on the header page.
PRINT-CMD-DEFAULTS	
LINE-SIZE	Number of characters per print line (default value) for the PRINTER operand of the PRINT command.
LOCK-FILE	Specifies whether the file to be printed is to be write-protected until the end of the spoolout job; default value for the LOCK operand of the PRINT command.
LEFT-MARGIN	Specifies the number of columns by which the output text is to be indented; default value for the SHIFT operand of the PRINT command.
DESTINATION	Destination of the spoolout job (local printer = computer center or remote printer)
DEVICE-NAME	Name of remote printer
POOL-NAME	Name of character set pool
ACC-EMPTY-F	Specifies whether empty files are to be accepted
ERROR-PROC (TRUNC)	Files which are to be deleted after printing are (not) deleted if an error occurs
FAMILY-PROC (LOCAL)	Default value for the FAMILY operand in the PRINT command for local jobs
FAMILY-PROC (REMOTE)	Default value for the FAMILY operand in the PRINT command for RSO jobs

# SHOW-SPOOL-PARAMETERS

Output field	Meaning
SPOOLOUT-OPTIONS	
ERASE-DISKET	Specifies whether a floppy disk is to be deleted before it is written to
SECT-TASK-LIMIT	Minimum number of area control tasks which may exist simultaneously
REM-DEV-LIM	Maximum number of RSO printers which may be active simultaneously
REM-BUF-SIZE	Buffer size in Kb (class 5 memory) for an RSO printer
MAX-MSG-TASK	Maximum number of simultaneously active message tasks; only relevant for RSO devices
RBP-STAT-LIMIT	Maximum number of simultaneously active remote batch terminals
ST-SP-LOW-VAL	Minimum value for the START-SPOOL=n operand of the PRINT/PUNCH command
FORM-NAME-LEN	The form section affected by scheduling
SYSLST-FORM	Standard form for outputs to SYSLST
SYSOUT-FORM	Standard form for outputs to SYSOUT
PRIORITY	
REPL-SESSION	Priority of the spoolout control task for output to replay tapes
FROM-POSITION	Priority of the spoolout control task if the FORM=mp operand in the PRINT command is specified
SECT-POSITION	Priority of area control task
DEVICE-TYPE	
ND-PRINTER	Specifies whether an ND printer is available
LP-PRINTER	Specifies whether an LP printer is available
HP-PRINTER	Specifies whether an HP printer is available
LP65-PRINTER	Specifies whether an LP65 printer is available
HP54-PRINTER	Specifies whether an HP54 printer is available
F-O-B	Maximum size of the FOB (number of sublines)
C-S-N	Maximum number of character sets
ROT	Specifies whether page rotation is possible
FO-OV	Specifies the electronic forms overlay which the printer can load
R-P-M	The memory containing forms overlays and character sets
CHECKPOINT-INTERVAL	
LI-PR	Spacing (in print pages) between the spoolout checkpoints for output to line printer (LI-PR), laser printer (PA-PR) and remote printer (REM-PR)
PA-PR	
REM-PR	
SPOOLOUT-SIZE	
UNIT	Unit (PAM pages or lines) specifying the size of the file to be printed; interrogated using the STATUS LIST command
LINES-FACTOR	Average number of records on one PAM page
MIN-L-P-P	Minimum number of lines on a print page

Output field	Meaning
SPOOLIN-OPTIONS	
LOG-DISKETTE	Specifies whether a log of the spoolin job is printed.
RSO-OPTIONS	The following fields designate default values for RSO operation
CONTR-LIMIT	Maximum number of RSO controllers for a system session
CONTR-DEV-LIM	Maximum number of RSO printers which can be operated by one controller
CHECK-PR-TIM	Time interval (in minutes) between 2 TIMEOUT checks on the printer
RETRY-TIME	Time interval (in seconds) between 2 attempts to send a message to a printer after an error
RELEAS-MEMORY	The DMS I/O areas are released if no further print job is pending (=YES) or the printer is deactivated (=NO)
CENTRAL-SCHEDULING	Scheduling criteria for the current local SPOOL session
DEVICE-WEIGHT	Weight (priority) for the following scheduling criteria for printer selection Range of values: (0,1,2,4,8,16,32,64,128)
ACC	Weight for the account number
C-S-N	Weight for the number of character sets required
CLASS	Weight for the spoolout class
FOB-S	Weight for FOB size
FORM	Weight for type of form
PRIO	Weight for spoolout priority
ROT	Weight for page rotation module
S-NAM	Weight for job name
US-ID	Weight for user ID
FO-OV	Weight for forms overlay
T-UP-P	Weight for "two-up" processing
JOB-SEQUENCE	Criteria used to sort spoolout jobs (distribution to queues); for example, FRM sorts according to form type, F-O according to whether or not an FOB is required
JOB-PRIORITY	Criteria for the order within a queue; for example, PRI = according to the priority of the spoolout job, AGE = according to how long the spoolout job has been in the system

## SHOW-SPOOL-PARAMETERS

---

Output field	Meaning
RSO-SCHEDULING	Scheduling criteria for the current RSO session
DEVICE-WEIGHT	Criteria for printer selection. Range of values: (0,1,2,4,8,16,32,64,128)
ACC	Weight for job number (accounting)
CLASS	Weight for ASPOOLOUT class
FORM	Weight for form type
PRIO	Weight for spoolout priority
S-NAM	Weight for job name
US-ID	Weight for user ID
JOB-SEQUENCE	Criteria used to sort the spoolout jobs (distribution to queues); for example, FRM specifies sorting by form type
JOB-PRIORITY	Criteria for the order within a queue; for example, PRI = according to the priority of the spoolout job, AGE = according to how long the spoolout job has been in the system
NEXT-CENTRAL- SCHEDULING	Scheduling criteria for the next local SPOOL session. Output fields as under CENTRAL-SCHEDULING.
NEXT-RSO-SCHEDULING	Scheduling criteria for the next RSO session. Output fields as under RSO-SCHEDULING.

## SHOW-TAPE-STATUS      Display tape allocation and monitoring

Application group: Interrogate current values (page 31)

### Command description

The SHOW-TAPE-STATUS command supplies information about the volume allocation type and state, the volume serial number, the device name, and the type of device allocation of the specified magnetic tapes. The tape can be identified via its VSN, an attribute or the device on which it is mounted. Information about monitored devices and those reserved "offline" is output separately. Only information about tapes reserved by the user's task is supplied.

### Format and operand description

Operation	Operands
<pre>{ SHOW-TAPE-STATUS } { SH-TAPE }</pre>	<pre>[ UNIT={ mn           (mn, ...) }   [ VOLUME={ vsn             (vsn, ...) } ]   ATTRIBUTE={ <u>ALL</u>               attribute } ]</pre>

**ATTRIBUTE** Restricts the information displayed to a subset of all tapes, i.e. those with the specified attribute. **ATTRIBUTE=...** must be specified in conjunction with **INFORMATION=STD**. This causes a record containing global allocation and monitoring information to be output for each tape of the subset.

**=ALL** A record is output for each tape with at least one of the attributes listed below; default value.

**=attribute** A record is output for each tape with the specified attribute. "attribute" may be any of the following:

Attribute	Meaning	
FREE EXCL[USIVE]}	} volume allocation state	
ONLINE MONITORED UNMONITORED PRE [MOUNT] MOUNT IN-USE		
CAN [CELLED] NO-DEV [ICE] REC [OVER] DISMOUNT POS [ITION]	} action state	
DMS SPECIAL WORK		
STD N[ON]-STD TAPE-MARK		
		} use mode
	} label type	

**UNIT** Specifies one or more tape devices via their mnemonic device names.

**=mn** Mnemonic device name.

**=(mn,...)** List of up to 26 mnemonic device names.

VOLUME	Indicates the tapes for which information is to be displayed. Wildcards may be substituted for any part at the beginning or end of the volume serial number to be specified. Information is displayed about all tapes whose VSN matches the specified substring. The asterisk ("*") should be used as the wildcard. Specification of the asterisk alone causes information about all tapes to be displayed that are monitored by NDM as well as those reserved "offline".
=vsn	Volume serial number of the tape (up to 6 characters).
=(vsn,..)	Up to 10 volume serial numbers can be specified in a list.

The output fields are described in more detail in the appendix.

## SHOW-USER-ATTRIBUTES Request information on user ID

Application group: Interrogation of current values (page 31)

### Command description

The SHOW-USER-ATTRIBUTES command allows you to request information about your user entries in the TSOSJOIN of the specified pubset. The output is directed to SYSOUT, SYSLST or SYSOUT and SYSLST.

### Format and operand description

Operation	Operands
$\left\{ \begin{array}{l} \text{SHOW-USER-} \\ \text{ATTRIBUTES} \\ \text{SH-U-A} \end{array} \right\}$	$\left[ \text{USER-IDENTIFICATION} = \left\{ \begin{array}{l} \text{*OWN} \\ \text{userid} \end{array} \right\} \right]$ $\left[ , \text{PUBLIC-VOLUME-SET} = \left\{ \begin{array}{l} \text{*HOME} \\ \text{*ALL} \\ \text{catid} \\ \text{(catid, ...)} \end{array} \right\} \right]$ $\left[ , \text{OUTPUT} = \left\{ \begin{array}{l} \text{SYSOUT} \\ \text{(SYSOUT, SYSLST (NUMBER=n25, PAGE-SIZE=n26))} \\ \text{SYSLST (NUMBER=n25, PAGE-SIZE=n26)} \\ \text{(SYSLST (NUMBER=n25, PAGE-SIZE=n26), SYSOUT)} \end{array} \right\} \right]$



- OUTPUT** Specifies where the JOIN entries are to be output.
- =SYSOUT** In interactive mode, output is directed to the terminal; in batch mode, to the printer.
  - =SYSLST** Outputs the information to printer in both batch and interactive mode.
- NUMBER=n25** Denotes the SYSLSTn file to which output is to be directed.  
Value:  $00 \leq n \leq 99$ ; default value: 00.
- PAGE-SIZE=n26**  
Determines the number of lines per page. Value:  $20 \leq n \leq 255$ ;  
default value: 64  
(see also class 2 system parameter PRPAPERL in the "System Installation" manual [13]).
- OUTPUT=(SYSLST(NUMBER=n25,PAGE-SIZE=n26),SYSOUT)**  
Controls output to SYSLST as described above; output to SYSOUT also takes place.
- PUBLIC-VOLUME-SET**  
Determines the pubset from which the user entries are to be output.
- =\*HOME** Outputs your JOIN entries on the home pubset (default value).
  - =\*ALL** Outputs your JOIN entries on all pubsets.
  - =catid** Outputs your JOIN entries on the specified pubset.
  - =(catid,...)** Your JOIN entries on up to 36 pubsets can be output.
- USER-IDENTIFICATION**
- =\*OWN** Outputs information on your user ID (default value).
  - =userid** Only your own user ID may be specified (1 to 8 characters).

## Alphabetical list of output fields:

### a) Output to SYSOUT

Output field	Meaning
ACCT-NB	Account number (8 characters)
ADDRSPACE	Maximum user address space in Mbytes (number between 0 and 2040).
AIDR,AIDW	Maximum values for test privilege level when testing\$ with AID (read, write); (number between 1 and 9).
AUDIT	Specifies whether the user may use the AUDIT function (YES or NO).
CLASS	Spoolout class (number between 0 and 255).
COMMANDS	User's command file name (54 characters or *NONE)
CPU-TIME	Maximum CPU time available for the account number displayed alongside (number between 0 and 65535).
CSTMP	Specifies whether you are authorized to use the CSTMP macro (YES/NO).
DEFCAT	Default catalog ID (1 character).
ENF	Indicates whether you may exceed the maximum public disk space allocated (YES/NO)
EXP	Permission for jobs to start immediately (YES/NO).
INHND	Task deactivation prohibition (YES/NO)
LIST OF ACCOUNT-NB	List with account numbers for the user ID
LIST OF JOB-CLASS	List of allocated job classes (8 characters).
MAIL-ADDRESS	Mailing address (64 characters or *NONE).
MAXAREC	Maximum number of user-specific accounting records (number between 0 and 32767).
MES-LANG	Language for message output (1 character)
MES-SEARCH	Scope applicable to search for messages in the task-specific language (TASK/ALL).
NTL	Permission for jobs with no time limitation (YES/NO).
PASS	Specifies whether the user is authorized to issue a PSWORD command (YES/NO).
PRI	Highest task scheduling priority for the user.
PSWORD	Specifies whether you are authorized to issue a PSWORD command (YES/NO/MOD).
RES-PAGES	Number of resident main memory pages (number between 0 and 2048).
SEVER	Specifies whether access to the system is locked for the user ID (YES/NO).
SPACE USED	Occupied public disk space in PAM pages (number between 0 and 16777215 or NOSPACE).
SPACELIMIT	Maximum public disk space available in PAM pages (number between 0 and 16777215).
TPIGNORE	IGNORE authorization for label checks in tape processing (YES/READ/BLP/ALL/NO).
TPRIV	Specifies whether you may use the debugging aid AID (YES/NO).
TTL	Specifies the permissible job types (STD/TP/SYS).
USERID	User ID (8 characters).

## b) Output to SYSLST

Output field	Meaning
ACCT-NB	Account number (8 characters).
ADSP	Maximum user address space in Mbytes (number between 1 and 2040).
AIDR, AIDW	Maximum value for test privilege level when testing with AID (read/write); (number between 1 and 9).
AUDI	Specifies whether you may use the AUDIT function (Y/N)
CLAS	Spoolout class (number between 0 and 255).
COMMANDS	Name of your command file (54 characters or *NONE).
CPUTIME	Maximum CPU time available for the account number displayed alongside (number between 0 and 65535).
CSMP	Specifies whether you may use the CSTMP macro (Y/N).
DCAT	Default catalog ID (1 character)
DMSGGS	Scope for message search in the language specified for the task (TASK/ALL).
DTL	Language for message output (1 character).
ENF	Specifies whether you may exceed the maximum public disk space allocated (Y/N).
EXPR	Permission for jobs to start immediately (Y/N).
INHDD	Task deactivation prohibition (Y/N)
JOB-CLASS	List of permitted job classes (8 characters)
ALLOWED	
MAIL	Mailing address (64 characters/*NONE).
MAX-ACC-REC	Maximum number of user-specific account records (number between 0 and 32767 or NL (No Limit)).
NTL	Permission for jobs with no time limit (YES or NO)
PASS	Specifies whether the user ID is password-protected (Y/N)
PRI	Highest task scheduling priority for the user.
PSW	Specifies whether the user may issue a PSWORD command (Y/N/M).
RES-PAGES	Maximum number of resident main memory pages (number between 0 and 2048).
SEV	Specifies whether access to the system is locked for the user ID (Y/N).
SPACE USED	Public disk space occupied by you in PAM pages (number between 0 and 16777215 or NOSPACE).
SPACELIMIT	Maximum public disk space available in PAM pages (number between 0 and 16777215).
TPIG	IGNORE authorization for label checks in tape processing (Y/R/B/A/N).
TPRI	Specifies whether you may use the debugging aid AID (Y/N).
TSK TYP LIM	Specifies the permissible task types (STD/TP/SYS).
USER-ID	User ID (8 characters).

Example

```
(IN)      SHOW-USER-ATTRIBUTES
(OUT)

USER-ID   :          USR123          SPACE-USED       :          0
GROUP-ID  :          *UNIVERSAL      SPACE-LIMIT      :          0
SEVER     :          NO              ADDRSPACE       :          64
PASS      :          YES             RES-PAGES       :          32767
PSWORD    :          MOD             MAXAREC        :          100
DEFCAT    :          W               MES-LANG       :
MES-SEARCH :          TASK          MES-LANG       :

CSTMP     : NO      AUDIT   : NO      ENF     : NO      TPIGNORE : NO
AIDRD     : 1      AIDWR   : 1      TPRIV  : YES

MAIL-ADDR : HR.MUSTERMANN          #44123 *45-123  9854      /PA 123
COMMANDS  : *NONE

+-----+-----+-----+-----+-----+-----+
!ACCT-NB !NTL!EXP!INH!TTL!PRI!CLASS! CPU-TIME !
+-----+-----+-----+-----+-----+-----+
!ACCXYZ12! NO! NO! NO!STD!210!      1! 2147480729!
+-----+-----+-----+-----+-----+-----+

LIST OF JOB-CLASS ALLOWED :
JCBMONCM JCBMONCP JCBMONDB JCBNACHT JCBQM315 JCB00050 JCB00200 JCB02000
JCB05000 JCB32000 JCDSTD
END OF DISPLAY FOR USER-ID = PA123456, PVS = :M :
```

## SKIP Branch according to task switch

Application group: Job control (page 22)

### Command description

The SKIP command enables you to have conditional and unconditional branches executed within a command file. Conditional branches are executed in accordance with the position of the specified task switches. If the specified switches are in the required position, a branch is made to the command with the label "name"; if no match is found, the next command is executed.

The SKIP command can be used in procedures and batch mode.

The branch is only performed if all specified switches are in the desired position (ANDed).

If the switch conditions are omitted in the SKIP command, control passes unconditionally to the command specified by "name".

In a procedure file (see the DO command), the branch can be either forwards or backwards. The system searches the file for the branch destination from the SKIP command to the end of file, and then resumes at the beginning of the file.

The period preceding the branch destination ".name" must not appear within a symbolic operand, since the branch destination as such would not then be found.

An error message, " .... SKIP TO NAME NOT FOUND", is displayed if the branch destination defined in the SKIP command does not exist.

### Format and operand description

Operation	Operands
SKIP	.name [, ON={no (no, ...)}] [, OFF={no (no, ...)}]

.name                    Label for a command to be executed if the values specified in the ON or OFF operand match the current switch setting. ".name" may be up to 8 characters long;  
Length of name ≤ 8 characters; see also page 7 and following pages.

no                        Number of the task switch whose position is to be checked;  
0 ≤ no ≤ 31. (The task switches are numbered from 0 to 31.)

- ON=no|(no,...) Specifies which switches are to be checked to determine whether they are set to ON.
- OFF=(no,...) Specifies which switches are to be checked to determine whether they are set to OFF.

### Examples:

#### *Example 1:*

Depending on the task switch setting, files X. are to be logically or physically deleted.

```
/PROC C
...
/SKIP .NULL,ON=(5)
/ERASE X.

/.ENDE ENDP
/.NULL ERASE X.,DESTROY
/SKIP .ENDE
```

If switch 5 is set to 0, no branch is made and the following log is produced:

```
(IN)      PROC C
          ...
(IN)      SKIP .NULL,ON=(5)
(IN)      ERASE X.

(IN)      .ENDE ENDP
```

If switch 5 is set to ON, by the command SETSW ON=(5) for example, the files whose names start with "X." are physically deleted (DESTROY operand in the ERASE command). The following log is generated:

```
(IN)      %/PROC C
          ...
(IN)      %/SKIP .NULL,ON=(5)
(IN)      %/.NULL ERASE X.,DESTROY
(IN)      %/SKIP .ENDE
(IN)      %/.ENDE ENDP
```

*Example 2:*

A branch list is set up in a procedure file by means of the SKIP command:

```

/PROCEDURE C, (&NAME=X)
/SKIP .&NAME
.....
/.X REMARK COMMAND-PART  X
.....
/SKIP .END
/.Y REMARK COMMAND-PART  Y
.....
/SKIP .END
/.Z REMARK COMMAND-PART  Z
.....
/.END ENDP

```

This allows you to choose, on calling the procedure file or during its execution, which of the three parts of this procedure file are to be executed.

Part Y is to be executed:

```
/DO PROC-FILE, (NAME=Y)
```

In interactive mode, this decision can be made during actual execution:

```

(IN)   DO PROC-FILE, (NAME=)
(IN)   %/PROCEDURE C, (&NAME=X)
(IN)   %/SKIP .&NAME
(OUT)  %&NAME=
(IN)   Y
:
:
```

## SKIPJV      Conditional branch in command sequence

Application group: Job variable functions (page 36)

The SKIPJV command is only available with the JV software product (see also "Job Variables" manual [11]).

### Command description

The SKIPJV command is used to perform conditional or unconditional branches within a command sequence.

### Format and operand description

Name	Operation	Operands
[.flag]	SKIPJV	[cond.exp], LABEL=.name

**cond.exp**                      May contain user and/or special job variables. If "cond.exp" is not present, an unconditional branch is made to ".name".

**LABEL=.name**

Specifies the branch destination, provided the condition specified in "cond.exp" is met. "name"="marker" is not permitted. If SKIPJV is issued within a procedure, then the command for the branch destination must also be entered within this procedure.

If the branch destination cannot be found, the ENDP command is executed in interactive mode, whereas in batch mode abnormal job termination will take place. If SKIPJV is issued in command mode, the branch destination command must not be included within a procedure. If it cannot be found, the job is terminated abnormally. If the branch destination command occurs in the ON/TIMEOUT statement sequence of an ON command, it will not be recognized.



## Execution logic of the SKIPJV command

- Batch processing mode:  
IF [cond.exp] = 'TRUE'  
    THEN
  - Branch to ".name" (forward or backward permissible); if ".name"=".marker", a branch to the next STEP command takes place, since a branch to the last command performed is not allowed.
  - Message on SYSOUT: 'SKIPJV: CONDITION = TRUE'
- ELSE
  - Continuation with the following command
  - Message on SYSOUT: '/SKIPJV: CONDITION=FALSE'
- Interactive mode:  
SKIPJV is only permitted in procedure mode; in all other cases the command is rejected with an error message.
- Procedure mode:  
As with batch processing mode. See also the explanation of LABEL=.name.
- Error exit:  
If the command cannot be executed, for example in the event of an error during the syntax check or when interpreting a conditional expression, a branch is made to the next STEP command.

For examples see the "Job Variables" manual [11].

## SKIPUS Branch according to user switch

Application group: job control (page 22)

### Command description

The SKIPUS command is used to perform conditional and unconditional branches in a command file. Conditional branches depend on the setting of the user switches (see the SETUS command). If the setting of the specified user switches matches the requested position, the command with the label ".name" is executed. If it does not match, the next command is executed.

The SKIPUS command can be used in procedures and in batch mode.

The user switches are stored in the JOIN file. Only the user switches in the JOIN file of the home pubset are used. User switches in imported JOIN files can be neither read nor modified.

The branch is only performed if all specified switches are in the required position (ANDed).

If the switch conditions are omitted in the SKIPUS command, control passes unconditionally to the command specified by "name".

In a procedure file (see the CALL command), the branch can be either forwards or backwards. The system searches the file for the branch destination from the SKIPUS command to end-of-file, and then from the beginning of the file.

The period preceding the branch destination ".name" must not appear within a symbolic operand, since the branch destination as such would not then be found.

An error message, " .... SKIP TO NAME NOT FOUND", is displayed if the branch destination defined in the SKIPUS command does not exist.

### Format and operand description

Operation	Operands
SKIPUS	.name [userid] [, ON= { no (no, ...) } ] [, OFF= { no (no, ...) } ]

.name	Label for a command to be executed in the event that the values specified in the ON and OFF operands match the current switch settings. Length of name $\leq 8$ characters; see also page 7 and following pages.
userid	User ID. The user ID must be specified if another user's switches are to be tested.
no	Number of the user switch whose position is to be checked. $0 \leq no \leq 31$ . (The user switches are numbered from 0 to 31.)
ON=no (no,...)	Specifies which switches are to be checked to determine whether they are set to ON.
OFF=no (no,...)	Specifies which switches must be checked to see whether they have been switched to OFF.

**Example:**

Command execution of the following procedure file is dependent on the setting of user switch 5.

```

/PROC C
.....
/SKIPUS .NULL,ON=(5)
/ERASE X.
/.ENDE ENDP
/.NULL ERASE X.,DESTROY
/SKIPUS .ENDE

```

} Procedure file SAL.PROC.SKIPUS

(01)

(02)

(01) Conditional branch to the label NULL if user switch 5 is ON.

(02) Unconditional branch to the label ENDE if no condition is specified.

The following log is created:

```

(IN)    %/PROC C
(IN)    %/SKIPUS .NULL,ON=(5)
(IN)    %/.NULL ERASE X.,DESTROY
(IN)    %/SKIPUS .ENDE
(IN)    %/.ENDE ENDP

```

If user switch 5 is set to ON, the files "X." are overwritten with zeros, i.e. the command ERASE X.,DESTROY is performed.

For a further example, see the SETUS command.

## SPARAM Compress SPOOL output

Application group: job control (page 22 ff.)

### command description

The SPARAM command can be used to influence the handling of feed control characters in edited files on output.

This command can be used to specify whether the feed control characters in the file are to be ignored (COMP=ALL) or interpreted differently (COMP=STD) on output. The option selected in the SPARAM command applies to all subsequent PRINT output requested with one of the operands SPACE=E, A or I. The option is cancelled when a SPARAM command without the COMP operand or with COMP=NO is specified or at the end of the current task.

### Format and operand description

Operation	Operands
SPARAM	[COMPRESS={ ALL STD NO}]

COMPRESS	Specifies how feed control characters are to be interpreted. The COMPRESS operand is only interpreted if the SPACE=E, A or I operand was specified in the PRINT command.
=ALL	Selects a one-line feed (minimum space requirement) for the output of edited files.
=STD	Interprets the carriage control characters in the edited file: <ul style="list-style-type: none"> <li>– Line feed:           1 line remains 1 line                           2 lines become 1 line                           3 lines become 2 lines.</li> <li>– All other line feeds become three-line feeds.</li> </ul>
=NO	Output of edited files will not be compressed.

## STAJV Output job variable attributes

Application group: Job variable functions (page 36)

The STAJV is only available with the JV software product (see also "Job Variables" manual [11]).

### Command description

The STAJV command returns information on the catalog entries of one or more job variables. The use of wildcards in the path name is permissible.

When entered without operands, STAJV causes a list of all job variables from the standard catalog of your user ID to be output.

The status display for temporary job variables always takes place under the internal name "S.TMP.nnnn.jvname", where "nnnn" is the task sequence number.

The 7-digit number preceding the displayed JV name indicates the length of the current JV value in characters (7 characters, as with PAM block output in the FSTAT command).

If STAJV addressed a currently active monitoring job variable, the line "JV-TYPE IS MONJV" is additionally output.

### Format and operand description

Operation	Operands
STAJV	$\left[ \begin{array}{l} \text{pathname} \\ *jvlink \\ \# \\ \$SYSJV. \end{array} \right] [ , ALL]$

- pathname** stands for: [:catid:][userid.][jvname]

"pathname" can be specified in fully or in partially qualified form. Wildcards may be specified for "catid", "userid" and "jvname".
- catid** Catalog ID of the pubset containing the job variable. Default value: the catalog ID (JOIN entry) assigned to the user ID.
- userid** User ID. Default value: user ID from the LOGON command. Specification of a user ID other than one's own causes the names of shareable job variables - and only these to be output.
- jvname** Name of the job variable. It can be specified in fully or partially qualified form. The names of temporary job variables must start with a prefix.
- \*jvlink** Specifies the job variable via its JV link name. The attributes of the associated job variables are output. The user of wildcards is not allowed with "\*jvlink".
- #** Outputs a list of the names of all temporary job variables. "#" stands for the prefix defined for temporary files and job variables by the system administrator.
- \$SYSJV.** Outputs a list of the names of all special job variables. The ALL operand is ignored.
- ALL** Causes all attributes of the specified job variable listed in the catalog to be output.

For examples see the "Job Variables" manual [11].

## STAM Read MRSCAT entries

Application group: Multiprocessor systems (page 37)

### Command description

Each pubset contains, among other things, the multiprocessor system catalog MRSCAT, which is a directory of all TSOS catalogs in the multiprocessor network. The STAM command can be used to output entries from the MRSCAT (MRS catalog) of the user's own pubset. The output is sent to SYSOUT. Refer to the "MCSF Multiprocessor System" manual [15] for further details.

### Format and operand description

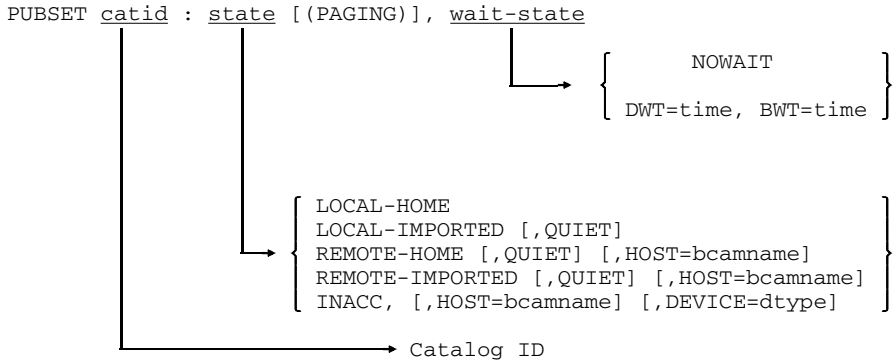
Operation	Operands
STAM	<pre>[ #   { wildcard }   { catid }    [ , SELECT= {     ALL     LOCAL     REMOTE     ACCESSIBLE     PAGING     SHARED     EXCLUSIVE     LOCAL-ACCESSIBLE     REMOTE-ACCESSIBLE   } ]</pre>

Default value:	All MRSCAT entries are to be output; preset value
#	Only the MRSCAT entry for the home subset is to be output.
wildcard	<p>The subsets whose catalog entries are to be output can be specified by means of wildcards, which include:</p> <ul style="list-style-type: none"><li>– "*" (substitute character for any character string, including blanks);</li><li>– "/" (substitute character for any single character);</li><li>– &lt;wildcard:wildcard&gt; (from ... to ...);</li><li>– &lt;wildcard,...&gt; (only those specified);</li><li>– "-" (all except those specified).</li></ul> <p>"wildcard" may be any string comprising one to four characters. All letters and digits are permitted for wildcards but no special characters.</p>
catid	<p>Only the MRSCAT entry of the specified subset is to be output. "catid" is a string consisting of one to 4 characters from the letters A...Z and the digits 0...9.</p> <p>The strings "PUB" and "PUBx" are not permitted.</p>
SELECT	Only permitted when selecting subsets with "_" or wildcards; limits the selection to specific subsets.
= <u>ALL</u>	No selection is made; default value.
=LOCAL	Only local subsets are to be selected.
=REMOTE	Only remote, i.e. non-local subsets are to be selected.
=ACCESSIBLE	Only accessible subsets are to be selected.
=LOCAL-ACCESSIBLE	Only local accessible subsets are to be selected.
=REMOTE-ACCESSIBLE	Only remote accessible subsets are to be selected.
=PAGING	Only paging subsets are to be selected.
=SHARED	Only shared subsets are to be selected.
=EXCLUSIVE	Only exclusive subsets (as opposed to shared subsets) are to be selected.



## Output format

- The following information is output for each selected entry:



A second line is output for shared pubsets that are local and not inaccessible:

```
SHARED, MASTER-HOST= OWN-HOST / bcamname
```

If a so-called "NON KEY PUBSET" is involved, it will be listed in an additional line if applicable.

When the pubset is not specified, output closes with the number of found entries:

```
1 ENTRY FOUND or
n ENTRIES FOUND
```

The individual specifications have the following meanings:

Output field	Meaning
PAGING	A paging area has been created on the public volume.
LOCAL	The catalog is accessible locally, i.e. it is managed by the processor on which the command was issued.
REMOTE	The catalog is not accessible locally, i.e. it is not managed on the processor on which the command was issued.
INACC	The catalog is inaccessible for MSCF. It may, however, be accessible via RFA, for example.
HOME	The catalog is a local or remote processor's own catalog.
IMPORTED	The catalog has been imported into a local or remote processor.
QUIET	The catalog is inaccessible (in the quiet state).
NO WAIT	A connection breakdown results in the state "inaccessible".
IN HOLD	The catalog connection is temporarily interrupted.
bcamname	BCAM name of the processor: <ul style="list-style-type: none"> <li>- on which the catalog is managed, or</li> <li>- on which the catalog was most recently managed, or</li> <li>- which was specified in the HOST operand.</li> </ul>
dtype	Device type (or unused).

# STATUS Request information on jobs and system load

Application groups:

Job control (page 22 ff.)

Interrogation of current values (page 31)

## Command description

The STATUS command provides you with a wide range of information with regard to the processing state of your own jobs as well as the load on the system.

Calling the command without specifying any operands is equivalent to the form STATUS TSN=tsn, where "tsn" is the TSN of the job in which the STATUS command was issued. The description of spoolout and RSO spoolout jobs is valid for SPOOL V2.5B und RSO V2.1B.

The STATUS command operands can be broken down into three groups, depending on the type of information supplied:

### 1. Information on the system load

Operand	Type of information
BIAS:	Number of resident memory pages currently allowed.
CATEGORY:	Load on task scheduler's queues (Q0-Q4,Q5,Q6) (task management)
JOB-CLASS:	Load on the job classes allowed for the caller (job management).
SUMMARY:	Number of jobs according to job type, e.g. spoolout jobs, interactive jobs, active batch jobs, etc.

- Information on all caller jobs or a subset of user jobs

Operand	Type of information	
SUMMARY:	Number of jobs according to job type.	
LIST:	Information on a designated group (subset) of jobs, e.g. name, TSN, job type, priority, CPU used, ... for the various jobs.	
ENVIR:	Hardware in the immediate environment of a designated group of jobs, e.g. station name, communication computer, catalog ID, output device, etc.	
PROG:	} { Information on a designated group of jobs of specific	
JOB:		} { job types, e.g. name, TSN, job type, size, ... for the
REPEAT:		} { individual jobs.

- Information on a caller's specific job

Operand	Type of information
TSN: NAME: JNAME: PNAME: MONJV: ]	[ Detailed information on the job designated by means of TSN or NAME or ..., e.g. TSN, job type, date and time, job name, priority, user ID, job class, LOGON time, size of print file, program size, task scheduling, etc. The output layout can be selected. It normally depends on the job type.

In the output information obtained from the STATUS command, the jobs are classified on the basis of job types as follows:

Job type (Tn/TYPE)	Explanation (definition)
1	Waiting batch job (in job pool)
2	Active batch job
3	(Active) interactive job
4	Waiting spoolout job
5	Active spoolout job
6	Waiting or active spoolout job from remote batch processing
7	Waiting, stopped (KEEP) or active RSO spoolout job
8	Active remote batch job

Task management subdivides tasks on the basis of task categories. Task categories play a part in task scheduling in accordance with the category concept (for details see the "System Administrator's Guide" manual [1]). The following (fixed) task categories exist:

SYS:            System and spoolout tasks  
 DIALOG:       Interactive tasks  
 BATCH:        Batch tasks  
 TP:            Transaction processing tasks

The system administrator is free to designate additional task categories.



In the interests of system performance, the task scheduler is not placed in a hold state while the STATUS command is being processed. As this may take some time to complete, the task situation could change in the interim period.

Format and operand description

Operation	Operands
<p>{STATUS}</p> <p>{STA }</p>	<p>BIAS</p> <p>CATEGORY</p> <p>JOB-CLASS [, STATE={ACT INACT HOLD}]</p> <p>REMOTE, DEVICE=printer</p> <p>SUMMARY [, {userid ALL}] [, TYPE={P S R}]</p> <p>[</p> <p>LIST [,userid] [, TYPE={P S n}] [, {</p> <p>IDENT={NONE UID JOB}</p> <p>INTYPE=( [min1] [, min2])</p> <p>TIMEREQ=( [sec1] [, sec2]) { [, ... ]}</p> <p>DEVICE=printer</p> <p>DESTINATION={printer pool}</p> <p>}]</p>

| Operation   | Operands   
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
---
---
---
--
--
---
---
---
---
{STATUS} {STA (cont.)
style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table>           ]           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> </tr> </table> </td> </tr> </table></td></tr></table></td></tr></table> | ENVIR[,userid][,TYPE= <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">P</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">S</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">n</td></tr> </table> ][, <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DEVICE=printer</td></tr> </table> [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DESTINATION=</td><td style="border: 1px solid black; padding: 2px;">printer</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">pool</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TERMINAL=</td><td style="border: 1px solid black; padding: 2px;">APPLICATION</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ORIGINAL</td></tr> </table> ]           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | P | S      | n    | IDENT= | NONE     |       | UID     |           | JOB   | INTYPE=( | [min1     | [,min2] | TIMEREQ=( | [sec1   | [,sec2]   | DEVICE=printer | DESTINATION= | printer |           | pool  
   | TERMINAL=   | APPLICATION |       | ORIGINAL | [         | ...  
   | ]   | <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">PROG[,userid][,TYPE=</td><td style="border: 1px solid black; padding: 2px;">P</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">2</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">3</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;"> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;"> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table>           ]           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> </tr> </table> </td> </tr> </table></td></tr></table> | [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">PROG[,userid][,TYPE=</td><td style="border: 1px solid black; padding: 2px;">P</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">2</td></tr>
<tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">3</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | PROG[,userid][,TYPE= | P   |        | 2       |        | 3    | IDENT= | NONE  |   | UID    |          | JOB   | INTYPE=( | [min1     | [,min2] | TIMEREQ=( | [sec1 | [,sec2] | [         | ...  
  | ]   | <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;"> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table>           ]           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> </tr> </table> </td> </tr> </table> | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | IDENT= | NONE  |        | UID  |          | JOB   | INTYPE=( | [min1     | [,min2] | TIMEREQ=( | [sec1 | [,sec2] | [ | ...   
   | ]   | <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table>           ]           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> </tr> </table> | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | IDENT= | NONE |  | UID |          | JOB   | INTYPE=( | [min1     | [,min2] | TIMEREQ=( | [sec1 | [,sec2] | [ | ...   | ]      | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table> ]           [ <table border="0" style="display: inline-table; vertical-align:
middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | IDENT= | NONE |  | UID |       | JOB  | DISP= | LIST  |   | ENVIR | [ | ... | ] |
| ENVIR[,userid][,TYPE= <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">P</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">S</td></tr> <tr><td style="border: 1px solid black; padding: 2px;">n</td></tr> </table> ][, <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DEVICE=printer</td></tr> </table> [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DESTINATION=</td><td style="border: 1px solid black; padding: 2px;">printer</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">pool</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TERMINAL=</td><td style="border: 1px solid black; padding: 2px;">APPLICATION</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ORIGINAL</td></tr> </table> ]           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | P  
  | S   | n | IDENT= | NONE |        | UID      |       | JOB     | INTYPE=(  | [min1 | [,min2]  | TIMEREQ=( | [sec1   | [,sec2]   | DEVICE=printer  | DESTINATION=  | printer        |              | pool    | TERMINAL= | APPLICATION   
   |   | ORIGINAL    | [     | ...      | ]         | <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">PROG[,userid][,TYPE=</td><td style="border: 1px solid black; padding: 2px;">P</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">2</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">3</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;"> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;"> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top;
padding-left: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table>           ]           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> </tr> </table> </td> </tr> </table></td></tr></table> | [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">PROG[,userid][,TYPE=</td><td style="border: 1px solid black; padding: 2px;">P</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">2</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">3</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | PROG[,userid][,TYPE=   | P   
   |                      | 2   |        | 3       | IDENT= | NONE |        | UID   |   | JOB    | INTYPE=( | [min1 | [,min2]  | TIMEREQ=( | [sec1   | [,sec2]   | [     | ...     | ]         | <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;"> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>   
       ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table>           ]           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> </tr> </table> </td> </tr> </table> | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | IDENT=  | NONE  |        | UID   |        | JOB  | INTYPE=( | [min1 | [,min2]  | TIMEREQ=( | [sec1   | [,sec2]   | [     | ...     | ] | <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border:
1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table>           ]           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> </tr> </table> | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | IDENT=  | NONE  |        | UID  |  | JOB | INTYPE=( | [min1 | [,min2]  | TIMEREQ=( | [sec1   | [,sec2]   | [     | ...     | ] | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table> ]           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | IDENT= | NONE   
  |        | UID  |  | JOB | DISP= | LIST |       | ENVIR | [ | ...   | ] |     |   |
| P   |  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| S   |  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| n   |  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| IDENT=  | NONE   
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | UID  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | JOB  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| INTYPE=(  | [min1  
  | [,min2]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| TIMEREQ=(   | [sec1  
  | [,sec2]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| DEVICE=printer  |  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| DESTINATION=  | printer  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | pool   
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| TERMINAL=   | APPLICATION  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | ORIGINAL   
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| [   | ...  
  | ]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">PROG[,userid][,TYPE=</td><td style="border: 1px solid black; padding: 2px;">P</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">2</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">3</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ]   | PROG[,userid][,TYPE=   
  | P   |   | 2      |      | 3      | IDENT=   | NONE  |         | UID       |       | JOB      | INTYPE=(  | [min1   | [,min2]   | TIMEREQ=(   | [sec1   | [,sec2]        | [            | ...     | ]         | <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px
solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;"> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table>           ]           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> </tr> </table> </td> </tr> </table> | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | IDENT=      | NONE  |          | UID       |   
  | JOB   | INTYPE=(   | [min1   
   | [,min2]              | TIMEREQ=(   | [sec1  | [,sec2] | [      | ...  | ]      | <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table>           ]           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> </tr> </table> | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | IDENT= | NONE     |       | UID      |           | JOB     | INTYPE=(  | [min1 | [,min2] | TIMEREQ=( | [sec1   
   | [,sec2]   | [   | ...   | ]      | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table> ]           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | IDENT= | NONE |          | UID   |          | JOB       | DISP=   | LIST      |       | ENVIR   | [ | ...  
  | ]   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |  
  |        |      |  |     |       |      |       |       |   |       |   |     |   |
| PROG[,userid][,TYPE=  | P  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | 2  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | 3  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| IDENT=  | NONE   
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | UID  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | JOB  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| INTYPE=(  | [min1  
  | [,min2]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| TIMEREQ=(   | [sec1  
  | [,sec2]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| [   | ...  
  | ]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ]   | IDENT=   
  | NONE  |   | UID    |      | JOB    | INTYPE=( | [min1 | [,min2] | TIMEREQ=( | [sec1 | [,sec2]  | [         | ...     | ]         | <table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top; padding-right: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table>           )           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table>           )           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> <td style="width: 50%; vertical-align: top; padding-left: 10px;">           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table>           ]           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table>           ]           [           <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table>           ]         </td> </tr> </table> | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | IDENT=         | NONE         |         | UID       |   
   | JOB   | INTYPE=(    | [min1 | [,min2]  | TIMEREQ=( | [sec1  
   | [,sec2]   | [  | ...  
  | ]                    | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table> ]           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ] | IDENT= | NONE    |        | UID  |        | JOB   | DISP=   | LIST   |          | ENVIR | [        | ...       | ]       |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| IDENT=  | NONE   
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | UID  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | JOB  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| INTYPE=(  | [min1  
  | [,min2]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| TIMEREQ=(   | [sec1  
  | [,sec2]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| [   | ...  
  | ]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">INTYPE=(</td><td style="border: 1px solid black; padding: 2px;">[min1</td><td style="border: 1px solid black; padding: 2px;">[,min2]</td></tr> </table> ) <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">TIMEREQ=(</td><td style="border: 1px solid black; padding: 2px;">[sec1</td><td style="border: 1px solid black; padding: 2px;">[,sec2]</td></tr> </table> )           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ]   | IDENT=   
  | NONE  |   | UID    |      | JOB    | INTYPE=( | [min1 | [,min2] | TIMEREQ=( | [sec1 | [,sec2]  | [         | ...     | ]         | ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">IDENT=</td><td style="border: 1px solid black; padding: 2px;">NONE</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">UID</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">JOB</td></tr> </table> ] <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">DISP=</td><td style="border: 1px solid black; padding: 2px;">LIST</td></tr> <tr><td style="border: 1px solid black; padding: 2px;"></td><td style="border: 1px solid black; padding: 2px;">ENVIR</td></tr> </table> ]           [ <table border="0" style="display: inline-table; vertical-align: middle;"> <tr><td style="border: 1px solid black; padding: 2px;">[</td><td style="border: 1px solid black; padding: 2px;">...</td><td style="border: 1px solid black; padding: 2px;">]</td></tr> </table> ]   | IDENT=  | NONE           |              | UID     |           | JOB   
   | DISP=   | LIST        |       | ENVIR    | [         | ...  
   | ]   |  | | | | | | | | | | | | | | | | | |
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| IDENT=  | NONE   
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | UID  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | JOB  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| INTYPE=(  | [min1  
  | [,min2]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| TIMEREQ=(   | [sec1  
  | [,sec2]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| [   | ...  
  | ]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| IDENT=  | NONE   
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | UID  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | JOB  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| DISP=   | LIST   
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
|   | ENVIR  
  |   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |
| [   | ...  
  | ]   |   |        |      |        |          |       |         |           |       |          |           |         |           |   |   |                |              |         |           |   
   |   |             |       |          |           |  
   |   |  |  
  |                      |   |        |         |        |      |        |   |   |        |          |       |          |           |         |           |       |         |           |  
  |   |   |   |        |   |        |      |          |       |          |           |         |           |       |         |   |   
   |   |   |   |        |      |  |     |          |       |          |           |         |           |       |         |   |   |        |   
   |        |      |  |     |       |      |       |       |   |       |   |     |   |

Operation	Operands				
<p>{STATUS}</p> <p>{STA</p> <p>(cont.)</p>	<table border="0" style="width: 100%;"> <tr> <td style="vertical-align: middle;"> <p>[ {TSN=}tsn NAME=name ]</p> <p>[ JNAME=jobname MONJV=jvname ]</p> </td> <td style="vertical-align: middle;"> <p>[ , DISP= { LIST ENVIR PROG ACT JOB REPEAT } ]</p> <p>TERMINAL= { APPLICATION ORIGINAL }</p> </td> <td style="vertical-align: middle;"> <p>[ IDENT= { NONE UID JOB } ]</p> </td> <td style="vertical-align: middle;"> <p>[ , ... ]</p> </td> </tr> </table>	<p>[ {TSN=}tsn NAME=name ]</p> <p>[ JNAME=jobname MONJV=jvname ]</p>	<p>[ , DISP= { LIST ENVIR PROG ACT JOB REPEAT } ]</p> <p>TERMINAL= { APPLICATION ORIGINAL }</p>	<p>[ IDENT= { NONE UID JOB } ]</p>	<p>[ , ... ]</p>
<p>[ {TSN=}tsn NAME=name ]</p> <p>[ JNAME=jobname MONJV=jvname ]</p>	<p>[ , DISP= { LIST ENVIR PROG ACT JOB REPEAT } ]</p> <p>TERMINAL= { APPLICATION ORIGINAL }</p>	<p>[ IDENT= { NONE UID JOB } ]</p>	<p>[ , ... ]</p>		

The operands are explained below in alphabetical order.

- ALL Provides information on the load on the system. The number of all jobs broken down according to job type is output.
- userid User ID of the caller. Information is output on all jobs running under the user ID.  
Default value: user ID from the LOGON command.
- BIAS Provides information on the maximum number of resident memory pages allowed at any given time.

Output field	Meaning
CORE	Number of main memory pages

**CATEGORY** Provides information on the load on the queues (Q0-Q4, Q5, Q6) managed by the task scheduler, broken down according to task categories (SYS, DIALOG, BATCH,...). (The output values provide a snapshot from task scheduling.)

Output field	Meaning
CATEGORY-NAME	Names of the task categories
MINMPL	Minimum multiprogramming level; minimum number of tasks that should be active in the category
MAXMPL	Maximum multiprogramming level; maximum number of tasks that should be active in the category
WEIGHT	Denotes the weight (urgency) for calculating activation priority.
#EXISTING	Number of tasks in this category
#ACTIVE	Number of tasks in queues Q0 - Q4 (ready active tasks).
#READY INACT	Number of tasks in queue Q5 (ready inactive tasks; these tasks are waiting for activation, i.e. main memory allocation).
#RDY NOT ADMITTED	Number of tasks in queue Q6 (ready inactive tasks not authorized for Q5; occurs when PCS is used).

**DESTINATION** Provides information on all spoolout jobs of the caller which are output on the designated RSO printers.

=printer Name of the RSO printer. The specification of wildcards is permissible; a subset of printers can thus be selected.

=pool Name of an RSO device pool. All printers in this pool can be addressed.



## DEVICE

Designates one or more RSO/RBP printers.

=printer

Name of the printer; the use of wildcards to designate a subset of printers is permissible.

Output field	Meaning
DEVICE	Name of the device (logical station name)
A	Availability. RBP devices: Y = Station is active N = No connection to station RSO printers: D = Not assigned (no SDVC command). A = Assigned but not active I = Inactive R = Active but not available S = Suspended (printing not possible)
STATION	Physical station name
PROCESS	BCAM processor name
USERID	User identification
TSN	TSN of the system device task
M	I = RBP terminal with spoolin devices O = RBP terminal with spoolout devices R = RSO printer
ERCOD	Error code for DCAM, PDN, printers
ERMSG	Error message code. The error message can be output by means of the HELP command
E	Exit routine activated (Y/N)
SRAM TSK	Number of the SRAM task managing the device
CTRL TSK	Number of the RSO task managing the device

- DISP                    Determines the output layout (if the standard layout for the current processing state is not to be displayed).
- =LIST                Output layout as specified in the LIST operand.
- =ENVIR              Output layout as specified in the ENVIR operand.
- =PROG               Output layout as specified in the PROG operand.
- =ACT                Provides information on task scheduling data.

Output field	Meaning
NAME	Job name, user ID or blanks, depending on the specification in the IDENT operand
TSN	Task sequence number
TID	Task identifier
UNP	UNPEND code with task scheduling
Q#	Number of the queue (task scheduling) in which the job is currently placed
SPOOLIN	Spoolin time (job acceptance)
LOGON	LOGON input time

- =JOB                Output layout as specified in the JOB operand.
- =REPEAT            Output format as specified in the REPEAT operand.

## ENVIR

Provides information on the peripheral hardware allocated to the caller's jobs. The output layout depends on the job type.

Output field	Meaning
NAME	Job name, user ID or blanks, depending on the specifications in the IDENT operand.
TSN	Task sequence number
STATION	Name (designation) of I/O device (terminal printer,...).
PROCESSOR	BCAM name of communication computer (front-end processor).
HOLD	Period for which the job is in the HOLD state.
MRCAT	Catalog ID and QUIET if the catalog is in the QUIET state, or catalog ID and HOLD if the catalog is in the HOLD state, or blanks (in all other cases).
FORM CLASS	Form name of paper used for printing. Spoolout class (as output in the JOIN entry; CLASS field).
DI	Forms overlay used for laser printer.
DEV	Output device for spoolout job (job types 4, 5 - see LIST operand)
PVS	Catalog ID of the pubset on which the file to be output is stored.
ERCOD	(BCAM) error code for an RSO spoolout job.
ERMSG	Error message number (only for an RSO spoolout job). The message text can be displayed by means of the HELP command.
OPT	The symbol "*" is output if more than 3 character sets, text overlaying (FOBs) or page rotation were specified for a spoolout job.

IDENT	Affects the entry for the NAME output field (the entry IDENT is only meaningful for the TSN, NAME, JNAME, PNAME and MONJV operands after DISP has first been specified).
=NONE	No entry.
=UID	The user ID is output.
= <u>JOB</u>	The job name is output; default value.
INTYPE	Makes it possible to limit the output to jobs (T1 - T3) with a specific processing time.
=(min1,min2)	Provides information on jobs which have been in the processing stage for a minimum of "min1" minutes and for a maximum of "min2" minutes. Default value: min1 = 0; min2 = 2147483647.
JNAME=jobname	Specifies the job name. Information is displayed on all batch and interactive jobs with the specified job name. If no name is specified, information is provided on jobs that are running under no specific job name. The output format depends on the specification for the DISP operand; otherwise, a special output format is required depending on the job type (see list below for output fields). Entering the IDENT operand is only meaningful in conjunction with DISP.

## JOB

Provides information on waiting and active batch and interactive jobs (job types 1, 2, 3).

Output field	Meaning
NAM	Job name, user ID or blanks, depending on the specifications in the IDENT operand.
TSN	Task sequence number.
TYPE	Job type and task category.
JCLASS	Job class.
INTYPE	Interval (in minutes) for which the job has been in the specified processing state.
P	Job priority.
START	Specifies the job start time in accordance with the LOGON or ENTER command, where: Eyyymmdd.hhmm= EARLIEST date.time Lyymmdd.hhmm= LATEST date.time Ayyymmdd.hhmm= AT date.time Whhmm= WITHIN interval BYOPER BYUSER SOON= SOON IMMED= IMMEDIATELY
REP	Job sequence (job repetition) as per command LOGON or ENTER, where: NO= No repetition STUP= AT-STREAM-STARTUP DAIL= DAILY WEEK= WEEKLY hhmm= PERIOD
RER	Specifies whether or not RERUN was specified in the LOGON or ENTER command (YES or NO)
FLU	Specifies whether FLUSH was specified in the LOGON or ENTER command (YES or NO).

## userid

Denotes the user ID. Information is output on all jobs running under the user ID. Default value: user ID from the LOGON command.

## JOB-CLASS

Provides information on the load on the job classes to which the user has access. Access can be limited to the active or inactive job classes or to those classes in the HOLD state.

Output field	Meaning
JCLASS	Names of job classes.
CLIM	Maximum number of jobs in the job class.
OPTM	Optimum number of jobs in the job class.
W	Weight (urgency) of the job class for job scheduling.
STATE	State of job class (ACT=active, INACT=inactive, HOLD=hold).
JSTREAM	Name of appropriate job stream.
DORM	Number of jobs waiting because the appropriate job scheduler is not yet active; or job run J(n), ( $n \geq 1$ ), of a repeat job which has not yet started.
ANCD	Number of jobs managed by the job scheduler.
WAIT	Number of jobs waiting for transfer to the job stream.
STRT	Number of jobs started (and still active).
HOLD	Number of jobs in hold state.

LIST

Provides information on all the caller's jobs. Output can be limited to a specific group of jobs, depending on the other operands specified (IDENT=/INTYPE=/TIMEREQ=).

Output field	Meaning
NAME	Job name, user ID or blanks, depending on the specification for the IDENT operand.
TSN	Task sequence number
TYPE	Job type
1	Batch job pending
DO	Dormant; the relevant job scheduler is not yet active; or job run J(i), (i≥1), of a repeat job.
WT	Waiting; the job is waiting to start
HO	Hold; job, job class or job stream is in the HOLD status
2/3	Active batch and interactive jobs
SYSTEM	Category for system tasks
BATCH	Category for batch jobs
DIALOG	Category for interactive jobs
TRANSAC	Category for transaction jobs
4	Spoolout jobs pending
SD	Printer with loadable VFB
SD7	Printer with loadable VFB
ND	Laser printer
NSD	ND + SD7 and/or SD
HP	High-performance laser printer
NHS	ND + HP + SD7 and/or SD
NHP	ND + HP
HSD	HP + SD7 and/or SD
PR	Any printer
FD	Floppy disk device
F70	3170 floppy disk device
F71	FD 3171 floppy disk device
T9N	Magnetic tape device with 800 bpi
T9P	Magnetic tape device with 1600 bpi
T9G	Magnetic tape device with 6250 bpi
TP	Any magnetic tape device.
WP	Waiting for preprocessing

Output field	Meaning	
	5	Active spoolout job
	mn	Mnemonic name of device on which job is currently being output Job reset by operator; classified as type 4 following release by the operator
	KP	
	6	RBP spoolout job
	WT	Waiting RBP spoolout job
	ACT	Active RBP spoolout job
	7	RSO spoolout job
	WT	Waiting RSO spoolout job
	ACT	Active RSO spoolout job
	KP	Reset RSO spoolout job
	WP	RSO spoolout job waiting for preprocessing
	PRE	RSO spoolout job being preprocessed
PRI	Job priority and run priority. The character "*" indicates an express job.	
CPU-USED	CPU time in seconds used by the job. Only specified for interactive jobs.	
CPU-MAX	Maximum CPU time in seconds available for the job HOLD The job has been halted by an NCHOLD (operator) command NTL TIME=NTL has been specified in the LOGON or ENTER command t The maximum CPU time available has been exceeded (t = time specified in the LOGON or ENTER command)	
ACCOUNT#	Account number (for job types 1/2/3 only).	



Output field	Meaning
SIZE	<p>Size of a file for output (only for job types 4, 5, 6, 7). The size is specified</p> <ul style="list-style-type: none"> <li>- in PAM pages if the parameter UNIT=PAM is displayed for SPOOLOUT-SIZE (see the SHOW-SPOOL-PARAMETERS command or</li> <li>- in print lines (printer), data records (floppy disk), PAM pages (magnetic tape) if UNIT=LINES is displayed. The value output is obtained from an estimate (depending on the operands FROM, TO, BINARY with STARTNO and ENDNO specified in the PRINT command)</li> <li>- in (logical) print pages if a system file is output. The number specified is suffixed with a letter P</li> </ul>
COPIES	Number of copies still to be printed
PRSIZE	<p>Number of lines (printer), data records (floppy disk) or PAM blocks (magnetic tape) already output at the time of the query; only if UNIT=LINES is displayed for SPOOLOUT-SIZE (and only for jobs of type 4, 5, 6, 7). If COPIES≠0 is specified, the value for PRSIZE is set to zero before each copy is output. The value for PRSIZE may differ from the calculated value for SIZE because:</p> <ul style="list-style-type: none"> <li>- The consequences, for example of a variable record length or specifying the SPACE=E operand can be included only approximately in the estimate.</li> <li>- When an interrupted spoolout job is restarted immediately, the output lines (records) are added to the previous value. The final value of PRSIZE may be greater than SIZE</li> <li>- When restarting after a wait state, the count recommences at zero. The final value of PRSIZE may be greater than SIZE</li> </ul>
RTSN	TSN of the job which generated the spoolout job (job types 4, 5, 6, 7 only).
OPT	The symbol "*" is output if more than 3 character sets, text overlay (FOB) or page rotation (ROTATION) were specified for a spoolout job.
DEVICE	Name of the device (printer) or RSO device pool (starting with an asterisk ("*")).

- MONJV=jvname** Specifies the name of the job variable monitoring the job. Information on this job is requested. The output layout depends on the specification for the DISP operand; otherwise, a special output layout is required depending on the job type (for output fields see list after operand description). Entering the IDENT operand is only meaningful in conjunction with DISP. This operand is only available with the JV software product .
- NAME=name** Specifies the job name. Information is requested for all batch, interactive and spoolout jobs with the specified job name. If no name is specified, information is provided on jobs that are running under no specific job name. The output layout depends on the specification for the DISP operand; otherwise, a special output format is required depending on the job type (for output fields see list following operand description). Entering the IDENT operand is only meaningful in conjunction with DISP.
- PNAME=pname** Specifies the job name. Information is requested for all spoolout jobs with the specified job name. If no name is specified, information is provided on jobs that are running under no specific job name. The output layout depends on the specification for the DISP operand; otherwise, a special output format is required depending on the job type (for output fields see list following operand description). Entering the IDENT operand is only meaningful in conjunction with DISP.

## PROG

Outputs information on active batch and interactive jobs (job types 2, 3). Included in the information is the name of the command currently being executed or the loaded application program, and the program size in PAM pages.

Output field	Meaning
NAME	Job name, user ID or blanks, as specified for the IDENT operand.
TSN	Task sequence number.
TYPE	Job type and task category.
SIZE	Number of virtual memory pages (class 6 memory) of 4 Kb each.
CURR-CMD	Command last executed (up to 8 characters) or HOLD if the job is in the HOLD state
PROGRAM-NAME	Name of loaded program (up to 64 characters)

## REMOTE

Provides information on device names, availability, terminal and processor names of the RSO/RBP printers specified under DEVICE=.

## REPEAT

Outputs information on job sequences (job repetition). (REPEAT operand in LOGON or ENTER command).

Output field	Meaning
NAME	Job name, user ID or blanks, depending on specification in the IDENT operand.
TSN	Task sequence number
TYPE	Job type and task category.
REP	Entries concerning job repetition, where: STUP= AT-STREAM-STARTUP DAIL= DAILY WEEK= WEEKLY hhmm= PERIOD
REPCNT	Specifies how many times the job will be started.
NTSN	Job number for next repetition.
NSTART	Calculated start time for next repetition. (Format: yymmdd.hhmm = date.time).

**STATE** Allows the output information to be restricted to active or inactive job classes or to job classes in the HOLD state. Information is provided on all job class states if the operand is not specified.

**=ACT** Outputs information on active job classes.


**=INACT** Outputs information on inactive job classes.

**=HOLD** Outputs information on job classes in the HOLD state.

**SUMMARY** Provides information on the number of jobs according to job types. Depending on the other operands specified, information is provided on the caller's jobs, the load on the system and selected job types. Job types T1 and T4 - T7 are only displayed in classified form if the operand TYPE=P/S/R has additionally been specified.

Output field	Meaning
T1/DO	Jobs waiting because the relevant job scheduler is not yet active; or job run J(i), (i≥1), of a repeat job.
T1/WT	Jobs waiting
T1/HO	Jobs held
T2	Active batch jobs
T3	Interactive jobs (active)
T4/PR	Waiting spoolout jobs (printer)
T4/FD	Waiting spoolout jobs (floppy disk)
T4/TP	Waiting spoolout jobs (tape)
T5/AC	Active spoolout jobs
T5/KP	Reset spoolout jobs which still be output in the same session.
T6/WT	Waiting RBP spoolout jobs
T6/AC	Active RBP spoolout jobs
T7/WT	Waiting RSO spoolout jobs
T7/AC	Active RSO spoolout jobs
T7/KP	Reset RSO spoolout jobs
T8	Active remote batch jobs and RSO devices

Each field consists of a number of up to 5 digits.

 The fields T4/PR to T8 are empty if SPOOL is not loaded.

TERMINAL	Specifies whether real terminal types and processor names or those specified in the DCAM application are to be displayed when working with DCAM applications (e.g. OMNIS) and DISP=E.
<u>=APPLICATION</u>	The terminal types and processor names defined by the application are to be displayed; default value.
=ORIGINAL	Real terminal types and processor names are to be displayed.
TIMEREQ	Makes it possible to limit the output to jobs (T1-T3) with a specific CPU time (time operand in LOGON or ENTER command).
=(sec1,sec2)	Provides information on jobs which have requested a minimum of "sec1" seconds and a maximum of "sec2" seconds of CPU time. Default value: sec1 = 0, sec2 = 2147483647.
TSN=tsn	Provides information on the job with the specified TSN. Default value: TSN of job in which the STATUS command is executed. The output format depends on the specification for the DISP operand; otherwise, a special output format is required depending on the job type (see list below for output fields). Entering the IDENT operand is only meaningful in conjunction with DISP.
TYPE	Makes it possible to limit output to specific job types. The admissible operand values in each case may be obtained from the format above (depending on which other operands are concurrently specified).
=P	Displays information on the number of jobs of types T1 - T3 (waiting and active batch and interactive jobs).
=S	Displays information on the number of jobs of types T4 - T8 (waiting and active spoolout jobs).
=R	Displays information on the number of jobs of types T6 - T8 (waiting and active RBP/RSO spoolout jobs and active remote batch jobs).
=n	Jobs of type "n" ( $1 \leq n \leq 7$ ).
=1	Waiting jobs (in the job pool).
=2	Active batch jobs.
=3	Interactive jobs.

**Alphabetical list of output fields on specification of  
TSN/JVNAME/PNAME/NAME/MONJV (if the DISP operand has not been specified).**

Output field	Meaning	Job type
ACCNB	Job account number (ENTER or LOGON command).	1,2,3
CHARS	Name of the character set specified in the PRINT command or character set pool, or name of the first character set of the specified list of character sets, or the number of a character set from the character set pool.	4,5
CHARS#	Number of character sets specified in the PRINT command.	4,5
CLASS	Job spoolout class (in accordance with the JOIN entry, CLASS field).	4,5,7
CMD	Last command received (up to 12 characters are output).	2,3
CONTROL	Indicates whether the CONTROL operand was specified in the PRINT command (PHYS/NO).	4,5
COPIES	Number of copies of the file still to be output and/or number of total page copies output.	4,5,6,7
CPU-MAX	Maximum CPU time available for job (in seconds)	1,2,3
CPU-USED	CPU time used by job (in seconds).	2,3
DEST	Name of the RSO device pool	4,5,7
DEVICE	Output device type.	4,5,6,7
DIA	Forms overlay used for laser printer.	4,5
ERCOD	(BCAM) Error code for RSO spoolout job.	7
ERMSG	Error message number. The message text can be requested with the HELP command.	7
FAMILY	Number of spoolout jobs if a partially qualified file name or several file names have been specified in the PRINT or PUNCH command.	4,5
FILENAME	File name of the file to be printed.	4,5,6,7
FLUSH	Indicates whether the FLUSH operand has been specified in the LOGON or ENTER command (YES or NO).	1,2
FOB	Name of the forms overlay buffer specified in the PRINT command.	4,5
FOBSIZE	Size of the forms overlay buffer specified in the PRINT command.	4,5
FORM	Form designation for paper used for printing.	4,5,6,7
HOLD	Period for which job is in the HOLD state.	2
INTYPE	Period (in minutes) for which a batch job is in the job pool.	1
JCLASS	Name of the job class.	1,2,3

Output field	Meaning	Job type
JOBNAME	Job name.	1,2,3
LOGON	Time of LOGON command input. Format: yymmdd.hhmm (=date.time).	2,3
MRSCAT	Catalog ID and state of the MRS catalog. (catid, QUIET/HOLD).	2
NOW	Time (date.time) on command input.	1,2,3,4,5,6,7
NSTART	Calculated start time for next job repetition (job sequence). Format: yymmdd.hhmm (=date.time).	1,2
NTSN	TSN for next job repetition (with job sequences).	1,2
PNAME	Spoolout job name.	4,5,6,7
POOL	Name of the character set pool specified in the PRINT command.	5
PRI	Job priority and run priority (run priority only for spoolout jobs).	1,2,3,4,5,6,7
PROC	Name (designation) of communication computer.	3
PROG	Name of loaded program (up to 54 characters).	2,3
PVS	Catalog ID of the pubset on which the file to be output is stored.	4,5,6,7
REPCNT	Indicates the number of times the job will have been started (job sequence).	1,2
REPEAT	Job sequence entries (in accordance with the LOGON or ENTER command). Possible entries are described in the REPEAT operand.	1,2
RERUN	Indicates whether the RERUN operand has been specified in the LOGON or ENTER command (YES or NO).	1,2
ROT	Page rotation specified in the PRINT command.	5
RTSN	TSN of the job that generated the spoolout job.	4,5,6,7
SIZE	Program size in PAM pages if a program is loaded.	2,3
SIZE	Size of the file to be output (see the LIST operand, SIZE output field). If several files are specified in the PRINT or PUNCH command, the total size is output. Zero is always specified for PLAM library members.	4
SIZE/PR	Size of the file to be output and of the print lines, data records etc. already output (see the LIST operand, SIZE and PRSIZE output fields).	5,6,7
SPOOLIN	Time (date.time-of-day) of job transfer to operating system (ENTER command, ..)	1,2

Output field	Meaning	Job type
START	Specifies job start (in accordance with LOGON or ENTER command).	1,2
STATION	Name (designation) of the I/O device (data display terminal, printer, etc.).	3
TID	Task identifier.	2,3
TSN	Task sequence number.	1,2,3,4,5,6,7
TYPE	Job type and task category.	1,2,3,4,5,6,7
UNP/Q#	UNPEND code and number of the job queue (task scheduling).	2,3
USERID	Caller's user ID.	1,2,3,4,5,6,7

**Determining the number of print lines (SIZE field) when a SAM-/ISAM file is output to printer:**

$$x = \begin{cases} \left. \begin{array}{l} b \\ - * d \\ r \end{array} \right\} & \text{for file with RECFORM = F and BLKSIZE} \leq 1 \text{ PAM page} \\ \left. \begin{array}{l} b \\ - * \frac{2048}{B} * d \\ r \end{array} \right\} & \text{for file with RECFORM = F and BLKSIZE} > 1 \text{ PAM page} \\ l * d & \text{for file with RECFORM = V} \end{cases}$$

where:

- x = number of print lines
- b = Size of a data block (BLKSIZE) in bytes
- r = Record length (RECSIZE) in bytes
- d = Size of file to be output in PAM pages (LASTPG field in catalog entry)
- l = LINES-FACTOR = average number of records per PAM page. The factor may be requested by means of the SHOW-SPOOL-PARAMETERS command; SPOOLOUT-SIZE output field.

The above equation applies only if the following are not specified in the PRINT command:

- STARTNO and/or ENDNO together with BINARY,
- FROM and/or TO.

The effect of these items is also taken into account when determining the print lines (maximum display = 999999).



**Examples:****1. Information on system loading**

```

(IN) STATUS BIAS _____ (01)
(OUT) CORE
      24 _____ (02)

(IN) STATUS JOB-CLASS _____ (03)
(OUT) JCLASS CLIM OPTM W STATE JSTREAM DORM ANCD WAIT STRT HOLD
      JCBMONCM 10 0 3 ACT JSMONSYS 0 0 0 0 0 0
      JCBMONCP 15 0 1 ACT JSMONSYS 0 0 0 0 0 0
      JCBMONDB 5 0 2 ACT JSMONSYS 0 0 0 3 0 0
      JCBNACHT 0 0 5 ACT JSSTD1 0 1 0 0 0 0
      JCB00050 10 0 6 ACT JSSTD1 0 0 0 1 0 0
      JCB00200 5 0 7 ACT JSSTD1 1 0 0 0 0 0
      JCB02000 5 0 8 ACT JSSTD1 3 1 0 0 0 0
      JCB05000 5 0 8 ACT JSSTD1 0 0 0 0 0 0
      JCB32000 2 0 9 ACT JSSTD1 0 0 0 0 0 0
      JCDSTD 130 0 2 ACT JSSTD1 0 0 0 77 0 0 } (04)

(IN) STATUS SUMMARY, ALL _____ (05)
(OUT) T1 T2 T3 T4PR T4PU T4FD T4TP T5AC T5KP T6 T7 T8
      10 16 81 208 0 0 0 1 0 0 1 11 (06)

```

- (01) The maximum number of resident memory pages allowed at any given time (for an application program) is to be output (important when entering **EXECUTE/LOAD**, CLASSII operand).
- (02) Up to 24 resident memory pages are allowed at this particular time.
- (03) The rate of utilization of those job classes allowed for the caller is to be output.
- (04) Only 1 batch job is being processed at the time of the request. Good prospects for a batch job being started rapidly.
- (05) The load on the system with respect to job types is requested.
- (06) 208 spoolout jobs are waiting for the printer. Poor prospects of a "rapid" listing.

2. Information on a group of jobs

```

(IN) STATUS LIST _____(01)
(OUT) NAME TSN TYPE PRI CPU-USED CPU-MAX ACCOUNT#
      ENTER1 MAA0 1 WT 9 210 0.0 22222 M4210MAN
      DIALOG2 M943 3 DIALOG 0 210 3.7280 9999 M4210MAN
      ENTER2 MA11 2 BATCH 9 215 0.1514 50 M4210MAN
      DIALOG1 M9L1 3 DIALOG 0 210 47.2070 9999 M4210MAN
(OUT) NAME TSN TYPE PRI SIZE COPIES PRSIZE RTSN OPT
      SPOOLOUT MA09 4 ND 210 1013 0 M9L1
(IN) STATUS LIST,TYPE=4 _____(03)
(OUT) NAME TSN TYPE PRI SIZE COPIES PRSIZE RTSN OPT
      SPOOLOUT MA09 4 ND 210 1013 0 M9L1
(IN) STATUS ENVIR _____(05)
(OUT) NAME TSN STATION PROCESSOR HOLD MRSCAT
      ENTER1 MAA0
      DIALOG2 M943 GB34842 DA1791
      ENTER2 MA11 BATCH NO
      DIALOG1 M9L1 GB34900 DA1791
(OUT) NAME TSN STATION FORM CLAS DI DEV PVS OPT
      SPOOLOUT MA09 STD 1 ND N
(IN) STATUS SUMMARY _____(07)
(OUT) T1 T2 T3 T4PR T4PU T4FD T4TP T5AC T5KP T6 T7 T8
      1 1 2 1 0 0 0 0 0 0 0 0 0 (08)

```

- (01) All of the caller's jobs are to be displayed. The job name is output in the NAME field (default value: IDENT=JOB).
- (02) 2 interactive jobs, 2 batch jobs and 1 spoolout job are recorded under the caller's user ID. Batch job ENTER1 has not yet been started. Spoolout job SPOOLOUT has been generated by the interactive job with TSN = 5681 and is still of job type 4 (waiting spoolout job).
- (03) All spoolout jobs of job type 4 are to be displayed.
- (04) 1 spoolout job is waiting for the printer; size of file to be output = 1013 PAM pages (output of the SHOW-SPOOL-PARAMETERS command indicated the UNIT=PAM parameter for SPOOLOUT-SIZE).
- (05) Information is to be output on the peripheral hardware allocated to the jobs.
- (06) Interactive jobs DIALOG1 and DIALOG2 are assigned to devices (data display terminals) GB34842 and GB34900. Batch jobs ENTER1, ENTER2 and spoolout job SPOOLOUT are (still) not assigned to any (external) devices. The file to be printed out is stored on the pubset with catalog ID N.

(07) The number of jobs according to job types is to be output.

(08) The following are recorded for the caller: 1 waiting and 1 active batch job, 2 interactive jobs, 1 waiting spoolout job.

3. Information on a specific job

```
(IN)  STATUS _____ (01)
(OUT) TSN:      5643      TYPE:    3 DIALOG  NOW:      870227.110512
      JOBNAME: DIALOG2   PRI:      0 210
      USERID:  PM123456  JCLASS:  JCDSTD  LOGON:    870227.1040
      ACCNB:   M1234MON  CPU-MAX:  9999   CPU-USED: 000004.1543
      STATION: GB34842   PROC:     DA1791
      TID:     000700CA  UNP/Q#:  00/000  CMD:      STATUS
```

```
(IN)  STATUS JNAME=DIALOG1 _____ (03)
(OUT) TSN:      5681      TYPE:    3 DIALOG  NOW:      870227.110551
      JOBNAME: DIALOG1   PRI:      0 210
      USERID:  PM123456  JCLASS:  JCDSTD  LOGON:    870227.1053
      ACCNB:   M1234MON  CPU-MAX:  9999   CPU-USED: 000047.8840
      STATION: GB34900   PROC:     DA1791
      TID:     0004006C  UNP/Q#:  17/012  CMD:      STATUS
      SIZE:    170      PROG:   :Z:$TSOS.EDT
```

```
(IN)  STATUS JNAME=ENTER2 _____ (05)
(OUT) TSN:      5711      TYPE:    2 BATCH   NOW:      870227.110608
      JOBNAME: ENTER2    PRI:      9 215   SPOOLIN: 870227.1104
      USERID:  PM123456  JCLASS:  JCB00050 LOGON:    870227.1104
      ACCNB:   M1234MON  CPU-MAX:  50     CPU-USED: 000000.1514
      REPEAT:  NO        RERUN:    NO     FLUSH:    NO
      MRSCAT:  NO        HOLD:     NO     START:   SOON
      TID:     000501CA  UNP/Q#:  17/012  CMD:      WAIT
```

```
(IN)  STATUS PNAME=SPOOLOUT _____ (07)
(OUT) TSN:      5709      TYPE:    4 ND      NOW:      870227.110626
      PNAME:   SPOOLOUT  PRI:      210    FAMILY:  0001
      USERID:  PM123456  FORM:     STD   SIZE:    1013
      DEVICE:  ND        CLASS:    0001  COPIES:  000/000
      RTSN:    5681      PVS:      N     DIA:
      DEST:    *CENTRAL  CONTROL:  NO
      FILENAME: MYFILE
      CHARS:
      CHARS#  FOB:
      FOB:      ROT:
      FOBSIZE:
```

```

(IN)  STATUS PNAME=SPOOLOUT _____ (09)
(OUT) TSN:      5709      TYPE:    5 L4      NOW:      870227.112436
      PNAME:    SPOOLOUT  PRI:      210      FAMILY:   0001
      USERID:  PM123456  FORM:    STD      SIZE/PR:  1013/014549
      DEVICE:   ND        CLASS:   0001    COPIES:   000/000
      RTSN:     5681      PVS:     N        DIA:
      DEST:     *CENTRAL  CONTROL: NO
      FILENAME: MYFILE
      CHARS:
      CHARS#:
      FOB:
      FOBSIZE:
      ROT:
  
```

```

(IN)  STATUS JNAME=DIALOG1,DISP=ACT _____ (11)
(OUT) NAME      TSN TID      UNP/Q#  SPOOLIN  LOGON
      DIALOG1  5681  00010162  17/12   870227.1053 910227.1053  (12)
  
```

- (01) Detailed information on the job in which the STATUS command has been executed is to be output. The command call is in the form of STATUS TSN=5643.
- (02) The STATUS command has been executed in interactive job DIALOG2 with TSN 5643. The caller is working at data display terminal GB34842. So far the job has used 4.1543 seconds of CPU time; run priority = 210; the LOGON command was entered at 10.40; ... . See the above list for the meaning of the output fields.
- (03) Detailed information on job DIALOG1 is to be output.
- (04) Data display terminal GB34900 is assigned to this interactive job. The last command to have been executed was the EXECUTE command; the TSOS.EDT program was started; program size 170 PAM pages. The program waits for an input (longer input), the task is in job queue 12 (long bourse wait) with UNPEND code 17 (remount after job queue 5), ... . See the above list for the meaning of the output fields.
- (05) Detailed information on job ENTER2 is to be output.
- (06) ENTER2 is an active batch job. Job class = JCB00050, task scheduling priority = 215. The maximum CPU time available is 50 seconds, 0.1514 seconds have so far been used. The last command executed was the WAIT command. The task is in job queue 12 (long bourse wait) with UNPEND code 17 (remount after job queue 5), ... . See the above list for the meaning of the output fields.
- (07) Detailed information on job SPOOLOUT is to be output.

- 
- (08) The spoolout job is waiting for the printer. A printer with the mnemonic name ND is allocated. The size of the file to be output is 1013 PAM pages; task scheduling priority = 210. Output is on standard paper (standard form, FORM = STD), ... . See the above list for the meaning of the output fields.
- (09) The SPOOLOUT job is requested once more (approx. 18 minutes later).
- (10) SPOOLOUT is now an active spoolout job. 14549 print lines had already been output at the time of the request. For conversion of PAM pages to print lines see page 632. For the meaning of the output fields see the list starting on page 630.
- (11) Data from task scheduling is to be output for the DIALOG1 job.
- (12) The task is in job queue 12 (long bourse wait) with UNPEND code 17 (remount after job queue 5). The internal task number is EB.

## STEP Set procedure section

Application group: Job control (page 22)

### Command description

The STEP command allows a command sequence to be divided into steps.

As a rule, an illegal command call in a procedure file or ENTER file activates a SPIN-OFF. This means that the commands following the illegal command are ignored, except for the STEP, ABEND, ABORT or LOGOFF commands. Whereas the ABEND or LOGOFF command terminates the job and the ABORT command terminates the procedure, the procedure continues normally from the STEP command. However, the STEP command must not begin with a symbolic name as a label.

The STEP command

- sets job switches 16 through 31 to off (see the SETSW command);
- resets the specifications for the language processors (see the PARAMETER command) to their default values;
- is not executed if a program is loaded. In this case, the procedure will be terminated and a corresponding error message issued.

A list of commands which do not trigger the SPIN-OFF mechanism if an error occurs can be found on page 438.

### Format

Operation	Operands
STEP	

**Examples:***Example 1:*

The following ENTER job is to catalog the file X.DAT. If the file is already cataloged, comments are displayed.

```

/LOGON...
/CAT X.DAT,SHARE=YES _____ (01)
.....
/SETSW ON=(6)
/STEP _____ (02)
/SKIP .FEHL,OFF=(6) _____ (03)
/LOGOFF
/.FEHL REMARK FILE X.DAT ALREADY EXISTS
/LOGOFF

```

- (01) If the file is already cataloged, the CATALOG command will not be executed in this form. SPIN-OFF is activated and execution branches to the next STEP command.
- (02) Command processing continues from the STEP command.
- (03) After the SKIP command is executed, a branch is made to the command labeled "FEHL", provided that switch 6 has not be set to ON, i.e. the SETSW ON=6 command has not been executed, file is already cataloged. No branch is made if job switch 6 has been set to ON (file could be cataloged).

*Example 2:*

The following commands are contained in a procedure file:

```

/PROC C
/CAN 1234
/REMARK NO SPIN-OFF
/SKIP .ENDE
/STEP
/REMARK SPIN-OFF
/.ENDE ENDP

```

The following log is produced upon execution of the procedure file:

```

(IN)      PROC C
(IN)      CAN 1234
(OUT)    % SCPO892 TSN NOT FOUND FOR USER. CMD TERMINATED.
(IN)      REMARK NO SPIN-OFF
(IN)      SKIP .ENDE
(IN)      .ENDE ENDP

```

The CANCEL command was rejected. However, there is no branch to the next STEP; instead, the REMARK command is executed.

For another example, see the DO and FILE commands.

## SYSFILE      **Assign system files**

Application groups:

File processing (page 26 ff.)

Program control (page 30)

### **Command description**

The SYSFILE command enables the user to alter the assignments of (system) files SYSDTA, SYSIPT, SYSLST, SYSLST01,..., SYSLST99, SYSOPT and SYSOUT (the last-named in batch mode only).

For (system) files SYSLST and SYSOPT, entries can be made for printer type and output format.

The SYSFILE command also enables the user to specify an object module file TASKLIB) for the Dynamic Linking Loader.

The (standard) file names SYSDTA, SYSIPT, SYSLST, SYSLST01, SYSLST02,..., SYSLST99, SYSOPT and SYSOUT denote (system) files used by the system for entering data or commands for the operating system, or for outputting data via the operating system. These files are created by the task involved, and initially refer (primarily) to predefined input or output areas.

As user you can cancel the primary assignment and assign your own (cataloged) files to these (standard) file names. Some of the standard names can also be made equivalent. The assigned file (to the right of the equal sign) then assumes the function of the (system) file (to the left of the equal sign).

When assigning an already cataloged file to a (system) file for output (SYSLST, SYSOUT, SYSOPT), the contents of the cataloged file are logically deleted (LASTPG = 00...0) -- unless the EXTEND operand is specified. Recovery of the file contents is possible, by using the DPAGE utility routine, for instance.

### **Files on floppy disk**

If a file extends over a number of floppy disks, it is only necessary to specify the volume serial number of the first floppy disk.

The file must be on the first volume specified in VOLUME.

Only one file can be assigned with the SYSFILE command. If you wish to assign a number of files, you must issue the SYSFILE command a corresponding number of times.

The SYSDTA system file is no longer assigned at procedure level if the end of the data inventory has been reached. Each subsequent read operation produces the message "SYSDTA NOT ASSIGNED". On returning to a lower procedure level, SYSDTA is



assigned to the device defined on this level.

If a read error occurs on floppy disk, the device is detached and no longer assigned to SYSDTA. Each further read task results in the message "SYSDTA NOT ASSIGNED".

If the system file SYSDTA is already assigned to a floppy disk, and there then follows a second assignment to the same device type and at the same procedure level, the command is not rejected, but any new values which may have been introduced are ignored.

The operator receives a message which contains a list of all volume serial numbers, and which requests him/her to mount the volume(s) concerned.

One of the following situations then applies:

- a) If the correct volume is mounted, the read process can be initiated and executed as normal.
- b) If the correct volume cannot be found, the operator responds with "NO", and sends an appropriate message to the caller.
  - If the problem arises upon entry of a SYSDTA command, the command is rejected and the SYSDTA assignment is not modified.
  - If the problem occurs during execution of the RDATA macro, the read process is terminated and the SYSDTA assignment no longer applies.
- c) If the operator enters a positive response, although (s)he has mounted the wrong volume, (s)he will receive a further message, together with a request to mount the correct volume. The operator then has the opportunity to repeat this process once more or to terminate it.
  - Access to the volume in INPUT mode: This is only possible if the following conditions relating to the VOL1 label are met:
    1. The access-indicator byte contains a blank.
    2. The owner identifier field consists of blanks.
    3. The owner identifier is identical to the user ID.

If the read process is carried out on a volume but the file name cannot be accessed, the RDATA macro is terminated with the message "ERROR DURING SWITCHING OF VOLUMES". The file name may be inaccessible because:

- the above-mentioned conditions (1 to 3) were not met;
- the command sequence was illegal;
- the track cannot be read;
- the volume cannot be mounted (etc.).

Unlike (system) files for output, assigned (cataloged) files generated by means of "SYSLST=filename" or "SYSOPT=filename" are not automatically printed out. You can have these files printed out by means of the PRINT command, or output to floppy disk by means of the PUNCH command.

The characteristics of the SYSDTA ... SYSOUT (system) files are explained on page 697 and subsequent pages ("BS2000 system files").

The table below shows the various formats for the SYSFILE command. The individual formats and their operands are described after the table.

**Formats and operand descriptions**

Operation	Operands
SYSFILE	$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{SYSDTA} \\ \text{SYSIPT} \end{array} \right\} = \left\{ \begin{array}{l} \text{pathname} \\ \text{\#filename} \\ \\ \text{(SYSCMD)} \\ \text{(PRIMARY)} \end{array} \right\} \\ \\ \text{SYSDTA}=\text{pathname1 (member)} [ , \text{VERSION}=\left\{ \begin{array}{l} \text{vers} \\ *STD \end{array} \right\} ] [ , \text{TYPE}=\left\{ \begin{array}{l} \text{type} \\ *STD \end{array} \right\} ] \end{array} \right\}$
	$\left\{ \begin{array}{l} \text{SYSLST} \\ \text{SYSOPT} \end{array} \right\} = \left\{ \begin{array}{l} \text{pathname} \\ \text{\#filename} \\ *DUMMY \\ \\ \text{(pathname, EXTEND)} \\ \text{(PRIMARY)} \end{array} \right\}$
	$\text{SYSLSTn} = \left\{ \begin{array}{l} \text{pathname} \\ *DUMMY \\ \text{(pathname, EXTEND)} \\ \text{(PRIMARY)} \\ *SYSLSTm \end{array} \right\}$
	$\text{SYSOUT} = \left\{ \begin{array}{l} \text{pathname} \\ \text{(pathname, EXTEND)} \\ \\ *DUMMY \\ \text{(PRIMARY)} \end{array} \right\}$
	$[\text{FILE} = \left\{ \begin{array}{l} \text{SYSLST} \\ \text{SYSOPT} \end{array} \right\}]$ $[ , \text{PRINTER} = \left\{ \begin{array}{l} 136 \\ 160 \end{array} \right\} [ , \text{HREC}=\text{m} ] [ , \text{FORM}=\text{code} ] [ , \text{LOOP}=\text{loop} ] ]$

Operation	Operands
SYSFILE (continued)	$[ , \text{COPIES} = \left\{ \begin{array}{l} \text{number1} \\ ([\text{number1}], \text{number2}) \end{array} \right\} ]$ $[ , \text{CHARS} = (z1 [ , z2 [ , z3 [ , z4 ] ] ) ]$ $[ , \text{CONTROL} = \left\{ \begin{array}{l} \text{PHYS} \\ \underline{\text{NO}} \end{array} \right\} [ , \text{IMAGE} = \text{xxxx} ] ]$ $[ , \text{SHIFT} = \text{columns} [ , \text{DIA} = \text{cc} ] ]$
	FILE=SYSOPT, DEVICE=DISKETTE
	$\text{TASKLIB} = \left\{ \begin{array}{l} \text{pathname} \\ (\text{NO}) \end{array} \right\}$
	$\text{SYSDTA} = \left( \left\{ \begin{array}{l} \text{DISKETTE} \\ \text{mn} \end{array} \right\} , \text{N[AME]} = \text{dsfnam} , \text{VOLUME} = \left\{ \begin{array}{l} \text{vsn} \\ (\text{vsn1}, \dots) \end{array} \right\} \right)$

If the parentheses are omitted when entering (SYSCMD), (PRIMARY) or (NO), these are interpreted as file names.

All data records read by SYSIPT are 80 bytes long. Longer records are truncated to 80 bytes and shorter records are padded with blanks to 80 bytes. An end-of-file condition is caused when reading in a command.

If SYSIPT is an ISAM file, a record with 80 bytes is provided. The 8-byte key is stored in positions 73 to 80 of the record. If records are shorter than 72 bytes, the record is filled with blanks to position 72 following the data.

SYSOUT can only be assigned in batch mode. A "/SYSFILE SYSOUT..." command entered in interactive mode is ignored (i.e. no SPIN-OFF is activated).

The command "SYSFILE FILE=SYSOUT" is not allowed. SPOOL operands can be entered via the PRINT command.

**Input from SYSDTA or SYSIPT**

Operation	Operands
SYSFILE	$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{[SYSDTA]} \\ \text{[SYSIPT]} \end{array} \right\} = \left\{ \begin{array}{l} \text{pathname} \\ \text{\#filename} \\ \text{(SYSCMD)} \\ \text{(( [PRIMARY] ) )} \end{array} \right\} \\ \\ \text{SYSDTA=pathname1 (member) , [VERSION= \left\{ \begin{array}{l} \text{[vers]} \\ \text{*STD} \end{array} \right\} ] [ , TYPE= \left\{ \begin{array}{l} \text{[type]} \\ \text{*STD} \end{array} \right\} ] \end{array} \right\}$

SYSDTA=

SYSIPT=

"pathname"

stands for [:catid:][userid.]filename

catid

Catalog ID of the pubset on which the file is stored.  
Default value: catalog ID assigned to the user ID (JOIN entry).

userid

User ID to which the file is assigned.  
Default value: user ID from the **LOGON** command.

filename

Name of a cataloged file (or file generation).

The file must be a SAM or ISAM file with variable-length records. For ISAM files, it is also necessary for the key to begin at byte 5 (KEYPOS=5) and to be 8 bytes long (KEYLEN=8). A file group can only be specified for tape files (not to be confused with file generation groups).

#filename

Name of a temporary file.

"#" is the character that was defined at system generation as a prefix for temporary files. Ask the system administrator for the precise character used.

"filename" is the name of any file (up to 30 characters).

The system can work without temporary files.

Temporary files are task-specific and are deleted when the task is finished.

There are 4 types of temporary files: BTAM, SAM, ISAM or PAM.

(SYSCMD)	The (system) file SYSCMD is made equivalent to the (system) file SYSDTA, i.e. SYSCMD additionally assumes the function of SYSDTA. It is then possible to read not only commands from SYSCMD but data as well.
([PRIMARY])	SYSDTA or SYSIPT is reset to the primary assignment. (The specification of an empty pair of brackets is also accepted).
SYSDTA	
=pathname1	stands for: [:catid:][userid.]library For "catid" and "userid", see SYSDTA=pathname.
library	Name of an LMS library (see the "LMS" manual [14]).
(member)	Name of a library member. The length of the expression "library(member)" must not exceed 42 characters minus the length of the catalog ID.
TYPE	Type of member (1 letter).
=type	Character from the set (D,S,M).
= <u>STD</u>	Default value: "type" = S.
VERSION	Supplements the member name with the version specification.
=vers	Version identifier (up to 10 characters).
= <u>STD</u>	Latest version (default value).

**Output to SYSLST or SYSOPT**

Operation	Operands
SYSDFILE	$\left. \begin{array}{l} \{ \text{SYSLST} \} \\ \{ \text{SYSOPT} \} \end{array} \right\} = \left\{ \begin{array}{l} \text{pathname} \\ \# \text{filename} \\ * \text{DUMMY} \\ (\text{pathname}, \text{EXTEND}) \\ ([ \text{PRIMARY} ]) \end{array} \right\}$

SYSLST=

SYSOPT=

pathname stands for: [:catid:][userid.]filename

catid Catalog ID of the pubset in which the file is stored.  
Default value: catalog ID assigned to the user ID (JOIN entry).

userid User ID to which the file is assigned.  
Default value: user ID from the **LOGON** command.

filename Name of a file or file generation.  
SYSLST or SYSOPT is assigned to this file or file generation. The file is created as a SAM file on public volume with the storage space allocation SPACE=(30,30).  
Alternatively, this file may also reside on private volumes, but you must specify this beforehand in a FILE command. Multifile tapes are not allowed.  
It is advisable to estimate the probable size of the file "filename" and to give a corresponding value for the SPACE operand in the FILE command so as to avoid too many memory requests.  
If no more memory space is available during SYSLST output to a disk file, the system requests a tape. The file is automatically copied to this tape and deleted from the disk. SYSLST output then continues to the tape file.

- #filename            Name of a temporary file.  
"#" is the character that was assigned at system generation time as a prefix for the file names of temporary files. Ask the system administrator for the precise character used.  
"filename" is the name of any file (up to 30 characters).  
  
The system can work without temporary files.  
  
Temporary files are task-specific and are deleted when the task is finished.  
  
There are four types of temporary files: BTAM, SAM, ISAM or PAM.
- \*DUMMY            A dummy file is assigned to SYSLST or SYSOPT. (For an explanation, see the FILE command; records are not stored.)
- (pathname,EXTEND)            The file "filename" is assigned to SYSLST or SYSOPT; records are inserted from the end of the file.
- ([PRIMARY])            SYSLST or SYSOPT is reset to its primary assignment. (The specification of an empty pair of brackets is also accepted.)

**Output to SYSLSTn**

Operation	Operands
SYSFILE	$\text{SYSLSTn} = \left\{ \begin{array}{l} \text{pathname} \\ *DUMMY \\ (\text{pathname}, \text{EXTEND}) \\ (\text{PRIMARY}) \\ *SYSLSTm \end{array} \right\}$

**SYSLSTn=** "n" is a 2-digit number from the set (01,02,...,99).  
 SYSLSTn files are only effective if (cataloged) SAM files are assigned to them. Their primary assignment is the file which is assigned to the (system) file SYSLST at the same moment.

**=pathname** stands for: [:catid:][userid.]filename

**catid** Catalog ID of the pubset on which the file is stored. Default value: catalog ID assigned to the user ID (JOIN entry).

**userid** User ID to which the file is assigned.  
 Default value: user ID from the **LOGON** command.

**filename** Name of a file or file generation.  
 The file is assigned to SYSLSTn and initialized as a SAM file on public volumes.  
 It is advisable to estimate the probable size of the file "filename" and to give a corresponding value for the SPACE operand in the FILE command.

**=\*DUMMY** A dummy file is assigned to SYSLSTn. (Records are not stored).

**=(pathname,EXTEND)**  
 The file "filename" is assigned to SYSLST; records are inserted from the end of the file.

**=(PRIMARY)** Primary assignment.



=\*SYSLSm "m" is a 2-digit number from the set (01,02,...,99), where  $n \neq m$ .  
The (system) files SYSLSm can also be assigned to each other.  
However, the following should be noted:

- Reciprocal assignment is not allowed, e.g.:

```
SYSFILE      SYSLSn  =  *SYSLSm
SYSFILE      SYSLSm  =  *SYSLSn
```

- The assignment must ultimately be to a cataloged or dummy file,  
e.g.:

```
SYSFILE      SYSLSn  =  filename
SYSFILE      SYSLSm  =  *SYSLSn
SYSFILE      SYSLSp  =  *SYSLSm
```

**Output to SYSOUT**

the SYSOUT operand is only permitted in batch mode.

Operation	Operands
SYSFILE	$\left. \begin{array}{l} \text{pathname} \\ (\text{pathname}, \text{EXTEND}) \\ *DUMMY \\ (\text{PRIMARY}) \end{array} \right\}$

**SYSOUT=**

**=pathname** stands for: [:catid:][userid.]filename

**catid** Catalog ID of the pubset on which the file is stored.  
Default value: catalog ID assigned to the user ID (JOIN entry).

**userid** User ID to which the file is assigned.  
Default value: user ID from the **LOGON** command.

**filename** Name of a file or file generation.  
The file is initialized as a SAM file on public volumes with the storage space allocation SPACE=(30,30) (see the FILE command). Alternatively, the file can also be on private volumes, but you must specify this beforehand in a FILE command.

It is advisable to estimate the probable size of the file "filename" and to give a corresponding value for the SPACE operand in the FILE command in order to avoid too many memory requests.

**(pathname,EXTEND)** The file "filename" is assigned to SYSOUT; data records are inserted from the end of the file.

**\*DUMMY** A dummy file is assigned to SYSOUT (data records are not stored).

**(PRIMARY)** SYSOUT is reset to its primary assignment.

Output from SYSLST or SYSOPT to printer

Operation	Operands
SYSFILE	<pre> FILE={   SYSLST   SYSOPT }  [, PRINTER={   136   160 }] [, HREC=m] [, FORM=code] [, LOOP=loop]  [, COPIES={   number1   ([number1], number2) }]  [, CHARS=(c1 [, c2] [, c3] [, c4))  [, CONTROL={   PHYS   NO }] [, IMAGE=xxxx]  [, SHIFT=columns] [, DIA=cc] </pre>

**FILE**  
 =SYSLST  
 SYSOPT

Describes the output form of the (system output) file SYSLST or SYSOPT, using the following operands. The most recent declarations prior to end-of-task always apply. FILE=SYSOPT must not be specified together with the COPIES operand.

**PRINTER**  
 =136  
 =160

No longer interpreted.

The operands HREC, FORM, LOOP, COPIES, CHARS, CONTROL, IMAGE, DIA and SHIFT determine the form of the printout; they are described under the PRINT command.

**Output from SYSOPT to floppy disk**

Operation	Operands
SYSFILE	FILE=SYSOPT, DEVICE=DISKETTE

FILE=.....            Outputs the (system) file SYSOPT to floppy disk under the name "OPT.tsn", where "tsn" is the task sequence number of the job. Output is governed by the default values (see the PUNCH command). If other output values are required, a cataloged file must be assigned to SYSOPT and output with the PUNCH command.

## Output for the Dynamic Linking Loader DLL

Operation	Operands
SYSFILE	TASKLIB={ pathname (NO)}

**TASKLIB** Denotes an object module file to be searched by DLL if

- an object module file (library) was not specified on loading a program and/or
- external references still have to be resolved

Default value: TASKLIB=(PRIMARY) i.e. the user file TASKLIB is searched or, if this does not exist, then the file \$TSOS.TASKLIB is searched.

=pathname stands for: [:catid:][userid.]filename

catid Catalog ID of the pubset on which the file is stored.  
Default value: catalog ID assigned to the user ID (JOIN entry).

userid User ID to which the file is assigned.  
Default value: user ID from the **LOGON** command.

filename Name of an object module file.  
When a program is linked, this file is searched for the object module file by the Dynamic Linking Loader (DLL), before the (user) file TASKLIB or (system) file \$TSOS.TASKLIB.

"filename" must not be the name of a file generation.

Nested procedures: following the ENDP command, TASKLIB is given the assignment which applied before the procedure was called.

=(NO) The assignment is cancelled. TASKLIB=(PRIMARY) applies or, for nested procedures, the assignment applies that existed before the procedure in question was called.

**Input from SYSDTA to floppy disk**

Operation	Operands
SYSFILE	$\text{SYSDTA} = \left( \begin{array}{l} \text{DISKETTE} \\ \text{mn} \end{array} \right), \text{NAME} = \text{dsfnam}, \text{VOLUME} = \left\{ \begin{array}{l} \text{vsn} \\ (\text{vsn}, \dots) \end{array} \right\}$

**SYSDTA=**

**=DISKETTE**      A floppy disk is assigned to SYSDTA.

**=mn**              Two-byte mnemonic device name of the floppy disk drive.

**NAME=dsfnam**    Name of the file set from the HDR1 label with a maximum length of 8 characters.

**VOLUME=**  $\left\{ \begin{array}{l} \text{vsn} \\ (\text{vsn}, \dots) \end{array} \right\}$

vsn = Volume serial number of the floppy disk (up to 6 characters).  
Up to 10 volume serial numbers may be specified.

**Examples:***Example 1: Changing the SYSDTA assignment*

```

/LOGON ...

/SYSFILE SYSDTA=X.PRIM ----- (01)
/EXEC $ASSEMB
/SYSFILE SYSDTA=(PRIMARY) ----- (02)
/EXEC $TSOSLNK
PROG X.LOAD
INCLUDE*      } Linkage editor assignments
END
/LOGOFF

```

- (01) At the beginning of the task, SYSDTA and SYSCMD are equivalent, i.e. they denote the same (primary) input source. The assembler always reads the source program from SYSDTA; the file X.PRIM containing the source program must therefore be assigned to SYSDTA.
- (02) After the assembly run, the linkage editor TSOSLNK is to be called. The linkage editor likewise reads statements from SYSDTA; these statements are entered at the terminal. SYSDTA must therefore be reset to its primary assignment.

*Example 2: Assigning a library member to SYSDTA*

The library LIB.TEST contains the member DATA with data for a program. This data is read in with the RDATA macro. SYSDTA must be assigned to this library member beforehand as follows:

```

/SYSFILE SYSDTA=BIB.TEST(DATA) ,TYPE=D

```

*Example 3: Changing the assignment of SYSLST*

For the following interactive job, certain sections of the printer listing are to be written to a cataloged file and the other sections are to be spooled out in the normal manner at the end of the job.

Fig. SYS-1          Assignment of SYSLST

The cataloged SAM file X.LISTS is not deleted when the job is finished. The following is an example of how the PRINT command might be used to print it out:

```
/PRINT X.LISTES, SPACE=E, COPIES=2
```



*Example 4: Assigning the TASKLIB*

Two object module files LIB.1 and LIB.3 were created using the LMR utility routine (see the "Utility Routines" manual [16]).

```
/SYSFILE TASKLIB=LIB.3
/EXEC (PROG22,LIB.1) _____ (01)
/SYSFILE TASKLIB=(NO)
```

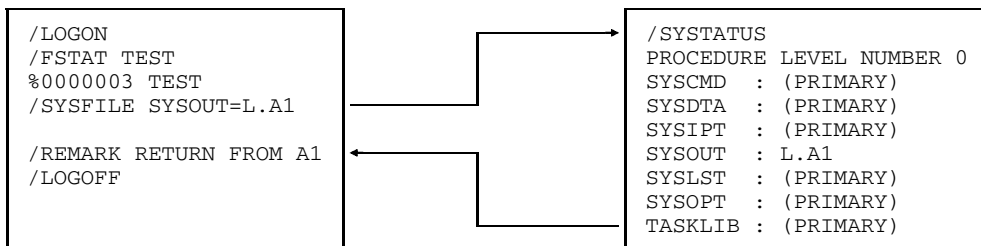
- (01) The EXEC command calls the Dynamic Linking Loader (DLL). DLL searches the LIB3 file for object modules to be linked to the PROG22 module from LIB1. If DLL cannot find the modules in LIB3, the search continues in the (user) file TASKLIB and then in the (system) file \$TSOS.TASKLIB.

*Example 5: Assigning SYSOUT*

```
An ENTER file contains the following commands: /LOGON
/FSTAT TEST
/SYSFILE SYSOUT=L.A1
/SYSTATUS
/SYSFILE SYSOUT=(PRIMARY)
/REMARK RETURN FROM A1
/LOGOFF
```

Runtime log SYSOUT

File L.A1



For further examples see the BREAK, LOAD, PARAMETER and SETUS commands.

## SYSTATUS Output information on system files

Application group: Interrogate current values (page 31)

### Command description

The SYSTATUS allows you to display the assignments (allocations) of the system files and TASKLIB object module file.

### Format and operand description

Operation	Operands
<pre>{SYSTATUS} {SYSTA }</pre>	<pre>[ ([SYSCMD] [, SYSDTA] [, SYSIPT] [, SYSOPT] [, SYSLST]   [, SYSLST01] [, SYSLST02] . . . [, SYSLST99] [, SYSOUT]   [, TASKLIB] ) ]</pre>

You can specify a number of system files, including TASKLIB. The parentheses may be omitted if only one system file is specified.

If no operands are entered, you will receive information on all system files, including TASKLIB.

The following output appears as a result of the **SYSTATUS** command:

```
PROCEDURE LEVEL NUMBER I
System file: Assignment
      .      .
      .      .
      .      .
```

The following assignments may be output:

Assignment	Meaning
filename	Refers to a system file assigned to a cataloged file. The prefix "\$userid" is additionally also output if this was specified in the SYSFILE command.
(PRIMARY)	For the predefined primary assignment (see page 697 and following pages)
(SYSCMD)	SYSDTA or SYSIPT is assigned to the system file SYSCMD.
(FLOPPY DISK)-MN=@@	SYSDTA is assigned to a floppy disk device. When the end of the file is reached, the device is released but the assignment is retained. In this case, "NOT ASSIGNED" is output.
NOT ASSIGNED	SYSIPT has not yet been assigned in interactive mode.



In ESCAPE mode, SYSCMD does not receive its primary assignment, but is assigned to the interrupted procedure file.

### Example:

Execution of the procedure file SAL.PROC.SYSTATUS was interrupted at procedure level 3. The user uses the STATUS command to requests information on the assignment of the various system files:

```
(IN)      REMARK SYSLST FILE ASSIGNED
(IN)      ER SAVLST.FEHL.
(OUT)     % EXC0015 ERROR IN PRECEDING COMMAND.
(OUT)     CMDS IGNORED UNTIL /LOGOFF OR /STEP INTERCEPTED
(IN)      STEP
(IN)      SYSTATUS
(OUT)     PROCEDURE LEVEL NUMBER 1
          SYSCMD : SAL.PROC.SYSTATUS
          SYSDTA : (PRIMARY)
          SYSIPT : NOT ASSIGNED
          SYSOUT : (PRIMARY)
          SYSLST : SAL.SYSTATUS
          SYSOPT : (PRIMARY)
          TASKLIB : (PRIMARY)
(IN)      ENDP
```

## TCHNG      **Change terminal characteristics**

Application group: Device control (page 33)

### **Command description**


The TCHNG command allows you to change the virtual characteristics of the terminal.


It can only be used in timesharing mode and is only effective in line or page terminal operation.


The default values for overflow control (OFLOW), operating mode (READ) and acknowledgment (TACK) are set by the system administrator at system generation time (for a detailed description see the "System Installation" manual [13]).

## Format and operand description

Operation	Operands
TCHNG	$[\text{OFLOW} = \left. \begin{array}{l} \text{NO} \\ \text{TIMER} \\ (\text{TIMER}, t) \\ \text{ACK} \end{array} \right\}]$ $[\text{, MAXLINE} = m]$ $[\text{, LINELEN} = n]$ $[\text{, LINEND} = \left. \begin{array}{l} \text{C 'NO' } \\ \text{C 'a' } \end{array} \right\}]$ $[\text{, READ} = \left. \begin{array}{l} \text{MODIF} \\ \text{UNPROT} \end{array} \right\}]$ $[\text{, TCHAR} = \left. \begin{array}{l} \text{LCASE} \\ \text{NLCASE} \\ \text{LHCOPY} \\ \text{CHCOPY} \\ \text{NHCOPY} \\ \text{GRAPH} \\ \text{NGRAPH} \\ \text{APL} \\ \text{NAPL} \end{array} \right\} [\text{, ...}]]$ $[\text{, SUB} = \left. \begin{array}{l} \text{C 'STD' } \\ \text{C 'a' } \end{array} \right\}]$ $[\text{, CORR} = \text{C 'a' }]$ $[\text{, TACK} = \left. \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}]$

CORR=C'a'	<p>"a" may be any character on the keyboard.</p> <p>When "a" is entered, the last character of the current line is erased ("backspace" function). The default for "a" is '_' (underscore). The CORR parameter is only evaluated for printer terminals, e.g. the 8103 Printer Terminal.</p>
LINELEN=n	<p>"n" specifies the line length of the terminal as a decimal number. For terminals with a physical end-of-line function (8103, 8110, 8161 in roll-up mode), dp system outputs are displayed using the line length specified by LINELEN. In all other cases, the output covers the entire actual line length, regardless of LINELEN. LINELEN is included in the calculations for the overflow control.</p> <p>Value of "n": <math>10 \leq n \leq 255</math>.</p>
LINEND	<p>Virtual end-of-line character for terminals whose hardware does not offer this function (8161, 8110).</p>
=C'NO'	<p>The virtual end-of-line is only characterized by a line change for computer output. For input, this function is not available.</p>
=C'a'	<p>"a" is any character entered from the keyboard to indicate the virtual end-of-line for output. Output continues in the next line. During input, the system transfers this character in the input text as NEW LINE to the user program.</p> <p>System default: "\" or "ö".)</p>
MAXLINE=m	<p>The preset overflow control action will be performed after "m" (decimal) output lines; <math>3 \leq m \leq 255</math>.</p>
OFLOW	<p>Specifies the type of overflow control desired. The system detects an imminent overflow whenever the next computer output (since the most recent terminal input) would exceed the number of lines specified for MAXLINE.</p>
=NO	<p>No overflow control.</p> <p>The system makes no provisions to allow the use of the ESCAPE function in the case of long outputs or to prevent premature overwriting of data on the terminal screen.</p>
	<p> In connection with OPTION MSG=T or the edit option OHCOPY=YES it is possible that only part of the output is reproduced on the hardcopy printer.</p>

- =TIMER/(TIMER,t)**  
 If an overflow occurs, the system waits "t" seconds in order for the terminal user to read the output data more easily and to initiate the ESCAPE/BREAK function.  
 "t" may be any of the decimal numbers 0,1...60. The default value is 6. A value of (TIMER,0) sets the wait time to 0. The TIMER value switches on the time overflow control without changing the most recently valid wait period.
-  The time entry is interpreted even if the output mode is changed (see also the CLEAR operand of the TCHNG macro in the "Executive Macros" manual [5]).
- =ACK**  
 Overflow control by acknowledgment.  
 If an overflow occurs, the system requests an acknowledgment from you by issuing the message "%PLEASE ACKNOWLEDGE". This allows you to control the speed of data output. Any inputs other than ESCAPE or BREAK may be used as acknowledgments. After receipt of an acknowledgment, dp system output is resumed.
- READ**  
 Controls the display of output messages on the screen and the type of input for 816x, 9749 and 975x Terminals.
- =MODIF**  
 Read modified fields.  
 The display of data on the screen is field-oriented; entry to the dp system is by transfer of modified fields.
- =UNPROT**  
 Read unprotected fields.  
 The display of data on the screen is without fields; entry is from the home position or the current cursor.  
 Entry from the home position is the default value. The F2 key can be used to switch from home position entry to entry from the cursor position (see also "Data Display Terminals" manual [18]).
- SUB**  
 Substitute for non-displayable characters. Illegal characters in output texts are replaced by the specified character.
- =C'STD'**  
 A device-specific smudge character is used as a substitute (default value).
- =C'a'**  
 "a" may be any keyboard character to be used as a substitute.

TACK	The system requests internal acknowledgment for output to terminal.
=YES	
=NO	The system does not request internal acknowledgment for output to terminal (not permitted for 8103 Printer Terminal).
	This setting may create resource bottlenecks in the system in the form of "unrecoverable errors". The current TACK setting applies for all subsequent output in program and system mode until reset by another TCHNG command or /LOGOFF (default value at system generation time).
TCHAR	Defines further terminal characteristics. If two or more values are entered, they must be enclosed in parentheses; if only one, the parentheses may be omitted.
=LCASE	Lowercase letters.
=NLCASE	Uppercase letters only.
=GRAPH	Graphics.
=NGRAPH	No graphics.
=APL	APL character set.
=NAPL	No APL character set.
=LHCOPY	Local hardcopy printer available directly at terminal.
	 The TCHAR=LHCOPY operand is evaluated only if a local hardcopy device was allocated when the connection was set up. Otherwise, it is ignored. If protected fields are used or OPTION OFLOW=NO, the screen contents will be incomplete.
=CHCOPY	Central hardcopy printer available at channel 0,...,31 of the same cluster controller to which the terminal is connected. The appropriate channel is the one specified in PDN when the terminal was generated (default value: 0).
=NHCOPY	No hardcopy printer.



**Examples:***Example 1:*

The following command is issued at the 8161 terminal:

```
(IN)      TCHNG OFLOW=TIMER,MAXLINE=3
(IN)      FSTAT X.
(OUT)     0000003 :V:$PA123456.X.ENTER.1
(OUT)     0000003 :V:$PA123456.X.ENTER.2
----- (01)
0000003 :V:$PA123456.X.ENTER.3
0000003 :V:$PA123456.X.ENTER.4
0000003 :V:$PA123456.X.ENTER.5
----- (01)
0000003 :V:$PA123456.X.ENTER.6
0000003 :V:$PA123456.X.ENTER.7
:V: PUBLIC:      7 FILES. RES=      21, FREE=      14, REL=      0 PAGES
```

- (01) At this point, the system waits for several seconds before continuing with the FSTAT output. During this delay, you can terminate the output by pressing the K2 key.

*Example 2:*

This command modifies the FSTAT output as follows:

```
(IN)      TCHNG OFLOW=ACK,MAXLINE=5,LINELEN=13
(IN)      FSTAT X.
(OUT)     0000003 :V:$PA123456.X.ENTER.1 ----- (02)
(OUT)     PLEASE ACKNOWLEDGE ----- (01)
0000003 :V:$PA123456.X.ENTER.2
(OUT)     PLEASE ACKNOWLEDGE
0000003 :V:$PA123456.X.ENTER.3
(OUT)     PLEASE ACKNOWLEDGE
0000003 :V:$PA123456.X.ENTER.4
(OUT)     PLEASE ACKNOWLEDGE
0000003 :V:$PA123456.X.ENTER.5
(OUT)     PLEASE ACKNOWLEDGE
0000003 :V:$PA123456.X.ENTER.6
(OUT)     PLEASE ACKNOWLEDGE
0000003 :V:$PA123456.X.ENTER.7
:V: PUBLIC:      7 FILES. RES=      21, FREE=      14, REL=      0 PAGES
```

- (01) The system waits until this message is acknowledged by pressing the DÜ (data transmission) key, after which output is continued.
- (02) Output is independent of LINELEN for the entire actual line length. However, LINELEN is used in calculating the overflow control.

## TRANSFER-FILE      Transfer file

Application group: File Transfer (FT) (page 37)

The TRANSFER-FILE command is only available with the FT software product.

### Command description

The TRANSFER-FILE allows you to transfer files between systems in an integrated computer network. The partner system (= remote system) can also operate with BS2000 or with one of the following operating systems: MSP (= BS3000), BS1000, MSDOS, IBM OS/VS2 MVS, SINIX, AMBOSS4.

The direction of transmission is selectable. The system where the command is input (= local system) may be either the sender or the recipient of the file.

This command is used to generate an independent job for data transfer.

In the TRANSFER-FILE command, you can specify a continuation command which is executed as a continuation job after file transfer, e.g. initiation of a (transferred) ENTER file.

In order to be able to access the user ID and the file on the partner system, you must specify the user ID, account number and passwords for this system - unless you are working with authorization profiles. The authorizations required for starting an FT job can be defined in an authorization profile (restriction to specific files, partner systems, security levels, restriction of continuation processing, definition of sender/recipient). This authorization profile is then referenced via the operand value \*NOT-SPECIFIED. The optional product FTAC-BS2000 is required if authorization profiles are to be used; cf. the "FTAC-BS2000" manual [26].

Default values **\*SAME** and **\*NONE** must not be specified explicitly for input.

The names of send and receive files written under a user ID other than that defined with the TRANS operand must be specified in the form \$userid.filename; otherwise specify "filename" only.

Filenames must not contain a catalog ID (catid).

Partially qualified file names and file generation group names are not allowed.

The library must be a PLAM library (library routine LMS, see the "LMS" manual [14]). This must be taken into account when specifying the member type. A library member may be transferred to an existing library as a member of it, or stored in the receiving system as a SAM file.

Commands must be specified with a slash; they may be up to 250 characters in length. Apostrophes which are part of the commands must be doubled. The total number of characters for the two commands in the send and receive section of the TRANSFER-FILE command must also not exceed 250.

Specifications for the remote system must conform to the conventions of that remote system.

Volumes:

FT-BS2000 accesses files which reside on public and/or private volumes (disks).

Certain conditions must be satisfied in order to process files on private volumes:

- You must catalog your file before transferring it (IMPORT function or the FILE and CATALOG commands).
- The private disk must have been mounted prior to file transfer.
- ISAM files must be entirely resident (indexes and data) on one volume.



This manual contains only an abbreviated description of the TRANSFER-FILE command (the description is valid for FT-BS2000 Version 4.0B). The "File Transfer" manual [7] contains a detailed description and application examples.

**Format and operand description** are subdivided into the following sections:

- Command section for the local system;
- Command section for the partner system (BS2000 operating system)
- Command section for the partner system (MSP operating system)
- Command section for the partner system (BS1000, MS-DOS, IBM OS/VS2 MVS, SINIX, AMBOSS4 operating systems)
- Command section for transfer attribute

## Command section for the local system

Operation	Operands
{TRANSFER-FILE} {TRANS-FILE {NCOPIY	TRANS={FROM TO }  ,PARTNER=name  ,LOC=  [(FILE={ *NOT-SPECIFIED filename *LIB({ *NOT-SPECIFIED } , { *NOT-SPECIFIED } ( { *STD } ) , { *NOT-SPECIFIED } ) ] ] library member version type ) ] ]  [, PASS={ *NONE password } ] *SECRET }  [, TRANS={ *SAME profile *SECRET (userid, accno [ , { *NONE password } ] ) ] } ] *SECRET }  [, PROC={ *SAME *NOT-SPECIFIED (userid, accno [ , { *NONE password } ] ) ] } ] *SECRET }  [, SUCC={ *NONE 'command' } ] [, FAIL={ *NONE 'command' } ]  [, LIST={ *SYSLST *LISTFILE } ] *NONE }  [, MONJV={ *NONE jvname } ]  [, JV-PASS={ *NONE password } ] ] *SECRET }

TRANS	Specifies the direction of transmission.
=TO	The file is transferred to the specified partner system.
=FROM	The file is called (copied) from the partner system and transferred to the local system.
PARTNER	Denotes the partner system.
=name	Specifies the symbolic name for the partner system. The designation is defined by the administrator responsible for file transfer and entered into the network reference documentation.
LOC=(...)	This operand configures the system. Subsequent specifications define the local system.
FILE	Denotes the file to be transmitted (TRANS=TO) or a name under which the file called from the partner system (TRANS= FROM) is to be cataloged within the local system.
= <u>NOT-SPECIFIED</u>	The file name has been defined in an authorization profile; default value.
=filename	Name of the file.
=*LIB(...)	The file is a member of a library.
<u>NOT-SPECIFIED</u>	The library name/member name/ member type has been defined in an authorization profile; default value.
library	Name of the library.
member	Name of the member.
version	Version name for the member.
*STD	Version entry (default value); highest version possible.
type	Type of member (1 letter).
PASS	Defines the file password.
= <u>NONE</u>	No password required; default value.
=password	File password, specified as a character string (C'....'), hexadecimal string (X'.....') or decimal digit.
=*SECRET	The password is requested; screen display of the password being input is suppressed.

TRANS={...}	Defines your authorizations for starting the data transfer job.
<u>*SAME</u>	The values (user ID, account number, password) from the current job are to be assumed; default value. Possible only if the user ID is not protected by a password.
profile	Authorization defined in an authorization profile, specified as a character string (C'...', 8-32 characters), hexadecimal string (X'.....', 8-32 characters) or alphanumeric character string (8-32 characters).
*SECRET	Your authorization is requested; screen display of the authorization being input is suppressed.
userid	User ID.
accno	Account number.
<u>*NONE</u>	The user ID is not protected by a password; default value.
password	Password protecting the user ID, specified in the form of a character string (C'...') or hexadecimal string (X'.....').
*SECRET	The password is requested; screen display of the password being input is suppressed.
PROC={...}	Denotes your specifications (authorizations) for starting a continuation job, e.g. starting an (ENTER) job.
<u>*SAME</u>	The entries for the TRANS operand also apply to PROC; default value.
*NOT-SPECIFIED	The authorizations for starting a continuation job have been defined in an authorization profile.
userid	User ID.
accno	Account number.
<u>*NONE</u>	The user ID is not protected by a password; default value.
password	Password for the user ID, specified in the form of a character string or hexadecimal string.
*SECRET	The password is requested; screen display of the password being input is suppressed.
SUCC=	Specifies a command to be executed in the form of a continuation job (only) after successful file transfer.
= <u>*NONE</u>	No continuation job; default value.

---

= 'command'	Name of a BS2000 command. The command must begin with a command slash (/).
FAIL=	Specifies a command which is to be executed as a continuation job (only) after unsuccessful file transfer.
= <u>NONE</u>	No continuation job; default value.
= 'command'	Name of a BS2000 command. The command must begin with a command slash (/).
LIST	Outputs a log relating to the data transfer.
= <u>SYSLST</u>	The log is written to (system) file SYSLST; this file is printed out when the job is finished; default value.
= <u>LISTFILE</u>	The log is written to a file named S.FT-BS2.no.LST, where "no" is the TSN or the FT job number. The file is stored under the ID specified with TRANS=....
= <u>NONE</u>	No log is printed.
MONJV	Specifies a job variable with which the FT job can be monitored.
= <u>NONE</u>	No monitoring by job variable.
= jvname	Name of the job variable.
JV-PASS	Specifies the password that has been defined for the job variable.
= <u>NONE</u>	No password has been defined.
= password	Password protecting the job variable, specified as a character string (C'....') or hexadecimal string (X'.....').
= <u>SECRET</u>	The password is requested; screen display of the password being input is suppressed.

## Command section for the partner system (BS2000 operating system)

Operation	Operands
{TRANSFER-FILE} {TRANS-FILE NCOPY (cont.)	<pre> [ , REM=<u>*BS2000</u>        { *SAME         *NOT-SPECIFIED         filename       }       ( [ FILE=<u>*LIB</u> ( { *SAME                        *NOT-SPECIFIED, { *SAME  *NOT-SPECIFIED } { *SAME  } } , { *SAME  *NOT-SPECIFIED } } ) ]        [ , PASS=<u>*NONE</u>         password       { *SECRET       } ]        [ , TRANS=<u>*SECRET</u>         (userid, accno [ , { *NONE                            password                            } ] )       { *SECRET       } ]        [ , PROC=<u>*NOT-SPECIFIED</u>         (userid, accno [ , { *NONE                            password                            } ] )       { *SECRET       } ]        [ , SUCC=<u>*NONE</u>         'command' ] [ , FAIL=<u>*NONE</u>         'command' ] ] </pre>

## REM

Specifies the operating system for the partner system.

=\*BS2000

The system runs under the BS2000 operating system; default value. All operand values must be specified using BS2000 syntax. The operand values are checked for syntax starting with the local system. Values specified for the local system are used if operands are omitted.

## FILE

Denotes the file to be transferred to the local system (TRANS=FROM) or a name under which the file sent from the local system (TRANS=TO) is cataloged.

=\*SAME

The entries from the local system are assumed; default value.



=*NOT-SPECIFIED	The file name has been defined in an authorization profile.
=filename	Name of the file.
=*LIB(...)	The file is a member of a library.
<u>*SAME</u>	The entries from the local system are used; default value.
library	Name of the library.
member	Name of the member.
version	Version designation for the member.
type	Type of member (1 letter).
*NOT-SPECIFIED	The library name/member name/ member type has been defined in an authorization profile.
PASS	Denotes the file password.
= <u>*SAME</u>	The same file password as for the local system is valid; default value.
=*NONE	No file password required.
=password	File password, specified as a character string, hexadecimal string or decimal number.
=*SECRET	The password is requested; screen display of the password being input is suppressed.
TRANS={...}	Defines your authorizations for working on the partner system.
<u>*SAME</u>	The values specified for the local system (user ID, account number, password) are assumed; default value.
profile	Authorization defined in an authorization profile, specified as a character string (C'....', 8-32 characters), hexadecimal string (X'.....', 8-32 characters) or alphanumeric character string (8-32 characters).
*SECRET	The authorization is requested; screen display of the authorization being input is suppressed.
userid	User ID.
accno	Account number.
<u>*NONE</u>	The user ID is not protected by a password.
password	Password protecting the user ID, specified in the form of a character string or hexadecimal string.

*SECRET	The password is requested; screen display of the password being input is suppressed.
PROC={...}	Defines your specifications (authorizations) for starting a continuation job, e.g. starting an (ENTER) job.
<u>*SAME</u>	The entries for the TRANS operand also apply to PROC; default value.
*NOT-SPECIFIED	The authorizations for starting a continuation job have been defined in an authorization profile.
userid	User ID.
accno	Account number.
<u>*NONE</u>	The user ID is not protected by a password; default value.
password	Password protecting the user ID, specified as a character string or hexadecimal string.
*SECRET	The password is requested; screen display of the password being input is suppressed.
SUCC	Specifies a command to be executed as a continuation job (only) after successful file transfer.
= <u>*NONE</u>	No continuation job; default value.
= 'command'	Name of a BS2000 command (including command slash).
FAIL	Specifies a command to be executed as a continuation job (only) after unsuccessful file transfer.
= <u>*NONE</u>	No continuation job; default value.
= 'command'	Name of a BS2000 command (including command slash).

## Command section for the partner system (MSP operating system)

Operation	Operands
{TRANSFER-FILE} {TRANS-FILE {NCOPI (cont.)	<pre>[ , REM=*MSP  (FILE={*NOT-SPECIFIED filename 'filename' }  [ , PASS={*NONE password } *SECRET }  , TRANS={ profile *SECRET (userid, accno[ , {*NONE password } ] ) *SECRET } }  [ , PROC={ *SAME *NOT-SPECIFIED (userid, accno[ , {*NONE password } ] ) *SECRET } } ]  [ , SUCC={*NONE 'command' } ] [ , FAIL={*NONE 'command' } ] ]</pre>

### REM

=\*MSP

Denotes the operating system of the partner system.

The system runs under the MSP operating system (previously known as BS3000). The operand values must be specified using MSP syntax, even though their syntax is checked beforehand by the local system.

**FILE** Specifies the file to be transferred to the local system (TRANS=FROM) or a name under which the file sent from the local system (TRANS=TO) is cataloged.

=\*NOT-SPECIFIED

The file name has been defined in an authorization profile; default value.

}

=filename

Name of the file. Can also be specified as a C string.

= 'filename'

**PASS** Defines the file password.

=\*NONE

No password required; default value.

=password

File password.

=\*SECRET

The password is requested; screen display of the password being input is suppressed.

**TRANS={...}**

Defines your authorizations for working on the partner system.

profile

Authorization defined in an authorization profile, specified as a character string (C'...', 8-32 characters), hexadecimal string (X'.....', 8-32 characters) or alphanumeric character string (8-32 characters).

\*SECRET

The authorization is requested; screen display of the authorization being input is suppressed.

userid

User ID.

accno

Account number.

\*NONE

The user ID is not protected by a password.

password

Password protecting the user ID.

\*SECRET

The password is requested; screen display of the password being input is suppressed.

**PROC={...}**

Defines your specifications (authorizations) for starting a continuation job, e.g. starting an (ENTER) job.

\*SAME

The entries for the TRANS operand also apply to PROC; default value.

---

*NOT-SPECIFIED	The authorizations for starting a continuation job have been defined in an authorization profile.
userid	User ID.
accno	Account number.
<u>*NONE</u>	The user ID is not protected by a password; default value.
password	Password protecting the user ID.
*SECRET	The password is requested; screen display of the password being input is suppressed.
SUCC	Specifies a command to be executed as a continuation job (only) after successful file transfer.
= <u>*NONE</u>	No continuation job; default value.
= 'command'	Name of an MSP command, specified as a character string.
FAIL	Specifies a command to be executed as a continuation job (only) after unsuccessful file transfer.
= <u>*NONE</u>	No continuation job; default value.
= 'command'	Name of an MSP command, specified as a character string.

## Command section for the partner system (BS1000, MS DOS, IBM OS/VS2 MVS, SINIX, AMBOSS4 operating systems)

Operation	Operands
{TRANSFER-FILE} {TRANS-FILE {NCOPI (cont.)	<pre>[ , REM=*ANY  (FILE= {   *NOT-SPECIFIED   'filename'   *LIB({     *NOT-SPECIFIED     'library'   }, {     *NOT-SPECIFIED     'member'   }, {     *NONE     *STD     'version'   }, {     *NONE     *NOT-SPECIFIED     'type'   }) }, [ , PASS={   *NONE   'password'   *SECRET }, [ , TRANS={   *NONE   profile   *SECRET   ('userid' [, {     *NONE     'accno'   }], [, {     *NONE     'password'   }]) }, [ , PROC={   *SAME   *NONE   ('userid' [, {     *NONE     'accno'   }], [, {     *NONE     'password'   }]) }, [ , SUCC={   *NONE   'command' }, [ , FAIL={   *NONE   'command' }] </pre>

### REM

Denotes the operating system of the partner system.

=\*ANY

The system runs under one of the operating systems BS1000, MS DOS, IBM OS/VS2 MVS, SINIX or AMBOSS4. The operand values must be enclosed in apostrophes and specified using the syntax of the remote system. Their syntax is not checked by the local system. Apostrophes which are part of the operand values must be doubled.

FILE	Specifies the file to be transferred to the local system (TRANS=FROM) or a name under which the file sent from the local system (TRANS=TO) is cataloged.
<u>=*NOT-SPECIFIED</u>	The file name has been defined in an authorization profile; default value.
= 'filename'	Name of the file.
=*LIB(...)	The file is a member of a library.
<u>*NOT-SPECIFIED</u>	The library name/member name/member type has been defined in an authorization profile; default value.
'library'	Name of the library.
'member'	Name of the member.
'version'	Version designation for the member.
*STD	Highest version level is used.
<u>*NONE</u>	No version specification; default value.
'type'	Type of member (1 letter).
<u>*NONE</u>	No version specification; default value.
PASS	Defines the file password.
<u>=*NONE</u>	No password required; default value.
= 'password'	File password.
=*SECRET	The password is requested; screen display of the password being input is suppressed.
TRANS={...}	Describes your authorizations for working on the partner system.
<u>*NONE</u>	The specifications are not required for the remote system; default value.
profile	Authorization defined in an authorization profile, specified as a character string (C'....', 8-32 characters), hexadecimal string (X'.....', 8-32 characters) or alphanumeric character string (8-32 characters).

*SECRET	The authorization is requested; screen display of the authorization being input is suppressed.
'userid'	User ID.
'accno'	Account number.
* <u>NONE</u>	Specification of an account number is not mandatory; default value.
'password'	Password protecting the user ID.
* <u>NONE</u>	The user ID is not protected by a password; default value.
*SECRET	The password is requested; screen display of the password being input is suppressed.
PROC={...}	Defines your specifications (authorizations) for starting a continuation job, e.g. starting an (ENTER) job.
* <u>SAME</u>	The specifications for the TRANS operand also apply to PROC; default value.
* <u>NONE</u>	The specifications are not required for the remote system.
'userid'	User ID.
'accno'	Account number.
* <u>NONE</u>	Specification of an account number is not mandatory; default value.
'password'	Password protecting the user ID.
* <u>NONE</u>	The user ID is not protected by a password; default value.
*SECRET	The password is requested; screen display of the password being input is suppressed.
SUCC	Specifies a command to be executed as a continuation job (only) after successful file transfer.
=* <u>NONE</u>	No continuation job; default value.
= 'command'	Name of a command.
FAIL	Specifies a command to be executed as a continuation job (only) after unsuccessful file transfer.
=* <u>NONE</u>	No continuation job; default value.
= 'command'	Name of a command.



## Command section for transfer attributes

Operation	Operands
{ TRANSFER-FILE TRANS-FILE NCOPY (cont.) }	[, COMP= { NONE BYTE } ]  [, WRITE= { REPLACE NEW EXT } ]  [, DATA= { *CHAR *BIN } ]  [, PRI= { *NORM *HIGH } ]  [, START= { *SOON *EARL ( ( { *TODAY *TOMOR yy-mm-dd } ) [ , { 00:00 hh:mm } ] ) ] } ]  [, CAN= { *NO *AT ( ( { *TODAY *TOMOR yy-mm-dd } ) [ , { 23:59 hh:mm } ] ) ] } ]

**CAN** Specifies a point in time after which the job can no longer be started (cancellation time).

=\*NO The job will be executed; default value.

=\*AT(...) The job will not be started after the specified time (date, time-of-day). The time must be in the future (22 days and 14 hours maximum) and must be after the time specified for START.

\*TODAY

The day on which the job is submitted; default value.

\*TOMOR

The day that follows the one on which the job is submitted.

yy-mm-dd Specifies a date in the form yy-mm-dd (yy=year, mm=month, dd=day). The hyphens must be specified.

23:59

Default value for time-of-day.

hh:mm

Specifies time-of-day in the form hh:mm (hh=hours, mm=minutes). The colon must be specified.

COMP	Specifies whether data of the send file are to be transferred in compressed or uncompressed form.
= <u>NONE</u>	Transfers data in uncompressed form; default value.
= <u>BYTE</u>	Transfers data in compressed form. (Compression involves successive bytes with the same contents).
DATA	Defines the data format of the send file.
= <u>*CHAR</u>	File contains text.
= <u>*BIN</u>	File contains data in binary form.
PRI	Defines the job priority to be used for starting the job, relative to other (transfer) jobs for the same remote system.
= <u>*NORM</u>	The job is given the standard priority for its job class.
= <u>*HIGH</u>	The job is given a higher priority, provided you have the appropriate authorization.
START	Defines a point in time for starting the job.
= <u>*SOON</u>	The job is to be started as soon as possible (taking into account its priority); default.
= <u>*EARL(...)</u>	The job is to be started no sooner than the specified time (date, time-of-day).
<u>*TODAY</u>	The day on which the job is submitted; default value.
<u>*TOMOR</u>	The day that follows the one on which the job is submitted.
yy-mm-dd	Specifies a date in the form yy-mm-dd (yy=year, mm=month, dd=day). The hyphens must be specified.
<u>00:00</u>	Specifies the starting time; default value.
hh:mm	Specifies the time-of-day in the form hh:mm (hh=hours, mm=minutes. The colon must be specified).
WRITE	Defines the OPEN mode of the receive file.
= <u>REPLACE</u>	If a file with the same name exists it is overwritten, otherwise a new file is created; default value.
= <u>NEW</u>	A new file is created. An existing file with the same name is not overwritten; in this case the send file is not transferred.
= <u>EXT</u>	If a file with the same name exists it is extended starting from end of file), otherwise a new file is created.

## TYPE Inform operator

Application group: Job control (page 22 ff.)

### Command description

The TYPE command can be used to send a message to a console.

It can be used in either interactive or batch mode.

Comments are not permissible in the TYPE command.

The message is usually sent to the central console. However, if it starts with "<" (less than), the next character is interpreted as a routing code (see the "System Operator's Guide" [2]) and the message is sent to the appropriate destination.

By specifying MSG=L in the LOGON or OPTION command, the user can have the console messages and operator responses for his job logged on SYSOUT.

### Format and operand description

Operation	Operands
TYPE	message

message                      The message is displayed to the operator and must not exceed 72 characters in length. All characters on the keyboard are permitted.

### Example

```
/TYPE *** LISTING FOR TSN 1433 WILL BE PICKED UP
```

The above message is displayed on the central console.

## VERIFY      Reconstruct file

Application group: File processing (page 26 ff.)

### Command description

The VERIFY command is used to make files (also file generations or file generation groups) accessible again which were abnormally terminated due to a system failure or job abortion.

The command can be used to:

- release a lock on a locked tape or disk file, thus making it available for general access again;
- reconstruct a disk file. For this purpose, the catalog entry is updated, the file is closed (if necessary) and, in the case of ISAM files, the file is reconstructed on the basis of the existing records.
- check an NK-ISAM file for consistency.

### Reconstructing ISAM files

- If the entry "pathname2" is omitted for an ISAM file on public volumes, the file is reconstructed on a work file provided by the system. "pathname1" is then deleted, without the DESTROY entry (see the ERASE command), and the work file is renamed "pathname1".
- If the entry "pathname2" is omitted for an ISAM file on private volumes, the file is reconstructed in a temporary work file on public volumes. The work file is subsequently copied to the file "pathname1", and the work file is deleted using the DESTROY entry (see the ERASE command). This procedure can be extremely time-consuming, and thus it is advisable to specify "pathname2".
- If "pathname2" is defined in the VERIFY command, ISAM file "pathname1" is reconstructed in this file, while "pathname1" itself remains unchanged. If "pathname2" is to be located on private volumes, or if "pathname1" is a file on private volumes, the user must catalog "pathname2" before issuing a VERIFY command. If the data and index blocks are located on separate volumes, the user must also allocate space for "pathname2" via a FILE command.

With ISAM files, data buffers are written back to the disks as soon as a new data block has to be loaded into main memory. This may result in the last changes being omitted from the reconstructed ISAM file. If, however, WROUT=YES was specified in the FILE command or macro, the error remains negligible, since the data buffer is written back to the disk after each change.

Data records in which keys and data are identical are only transferred to the reconstructed file once if they are in various data blocks.

No space is reserved in the data blocks of the reconstructed file for subsequent additions; this corresponds to PAD=0 in the FILE command.

ISAM files whose data and index blocks are on separate private volumes can only be reconstructed with the VERIFY command when BLKSIZE=STD.

If an ISAM data block contains data which cannot be assigned to any defined record, the whole block is saved to a PAM file called "S.filename1.REPAIR". After termination of the VERIFY processing, this file is made available to you for your own attempts at reconstruction. If the file name becomes too long, "filename1" will be shortened accordingly.

Since a copy of the file is made when ISAM files are reconstructed, and this copy is accommodated in public space, care must be taken that enough public space is available. If "pathname1" is a temporary file, it is advisable to specify "pathname2" as a temporary file as well.

Following the "FILE NOT ACCESSIBLE DUE TO SPACE PROBLEM" message, the file can no longer be made available by means of VERIFY. In this case, only the command "/ERASE\_...,CATALOG" can be given (also applies to public files).

For files declared as exclusively reserved via the SECURE-RESOURCE-ALLOCATION command, you can release the lock; in the case of disk files, only the system administrator is authorized to do so.

If file access was interrupted while data buffers were in main memory, the last few changes may be missing from the reconstructed file. This is because buffers are not written to the external storage until they are full.

**Format and operand description**

Operation	Operands
VERIFY	<p>pathname1 [, pathname2]</p> <p>[ , REPAIR= {                    YES                    ABS                    NO                    CHECK          } ]</p> <p>[ , SUPPORT= {                    PUBLIC                    PRDISC          } ]</p>

pathname1 stands for: [:catid:][userid.]filename1 and specifies the file (permanent or temporary file), file generation group or file generations to be reconstructed.

If "pathname1" is specified in partially qualified form, then

- "pathname2" (if required) must also be specified in partially qualified form;
- "pathname2" will be ignored when reconstructing file generations;
- only files not properly closed will be taken into account for REPAIR=NO;
- REPAIR=ABS is prohibited.

pathname2 stands for: [:catid:][userid.]filename2 It is only relevant for the reconstruction of ISAM files and specifies the file on which the ISAM file "pathname1" is to be reconstructed.

"pathname2" must be different from "pathname1"; if "pathname1" is specified in partially qualified form, so must "pathname2".

"pathname1" may designate a permanent or temporary file, but not a file generation or file generation group.

If "pathname2" is omitted, a work file is created by the system for the reconstruction of the ISAM file "pathname1". "pathname2" must be specified, however, if the ISAM file is stored on private disk, and the index section and the data section are stored on different disks.

catid	Catalog ID of the pubset on which the file is stored. Default value: catalog ID assigned to the user ID (JOIN entry).
userid	User ID to which the file is assigned. Default value: user ID from the LOGON command.
filename1	Fully or partially qualified file name, name of a file generation, a file generation group or temporary file.
filename2	Name of the file on which the ISAM file "filename1" is to be reconstructed. If this entry is omitted, the ISAM file will be reconstructed on a work file created by the system. An ISAM file on private volumes for which data and index blocks are on different disks will only be repaired if "filename2" is specified. "filename2" must be different from "filename1" and must be a fully or partially qualified file name, i.e. the entry will be ignored if it is the name of a file generation or a file generation group.  Temporary ISAM files may be specified.  If a foreign user ID is specified in a path name, the file must have been made shareable.
REPAIR	Specifies the type of file reconstruction to be used for the files referred to by "pathname1". In the event of tape files, only REPAIR=NO is permitted.
<u>=YES</u>	<p>PAM: The file is unlocked. If the file is identified as open, it is closed. The last-page pointer is set to the highest possible value (i.e. for BLKCTRL=PAMKEY, to the file size [FILESIZE], and for BLKCTRL=DATA/NO to the file size rounded up to a multiple of the block size).</p> <p>SAM: The file is unlocked. If the file is identified as open, the last-page pointer is set to the last PAM page written, and the file is closed. If the file is on a mirror disk (see the "DRV" manual), equivalence between the block contents is restored if needed.</p> <p>ISAM: The file is unlocked and reconstructed. The defined secondary keys are recreated; incomplete secondary keys are ignored.</p>

- =ABS
- PAM: The file is unlocked.  
 If the file is identified as being open, the last-page pointer is then updated so as to set it to the last PAM page written in the case of files with BLKCTRL=PAMKEY/DATA or to the highest value (file size rounded to a multiple of the block size) for files with BLKCTRL=NO.  
 If the file is on a mirror disk (see the "DRV" manual), equivalence between block contents is restored if needed.  
 If the file is not identified as open, last-page pointer (LPP value) remains the same.
- SAM: The file is unlocked.  
 The last-page pointer is set to the last PAM page written, and the file is closed. If the file is on a mirror disk (see the "DRV" manual), equivalence between block contents is restored if required.
- ISAM: The file is unlocked and reconstructed. The defined secondary keys are recreated; incomplete secondary keys are ignored.
- =NO
- PAM: The file is only unlocked (i.e. the entry in the table of locked files is removed) and continues to remain open under certain circumstances, i.e. for FSTATUS STATE=NOCLOSE it will still be listed, whereas for VERIFY ..., REPAIR=YES it is considered to be a file in need of repair.
- SAM: The file is only unlocked (i.e. the entry in the table of locked files is removed) and continues to remain open under certain circumstances, i.e. for FSTATUS STATE=NOCLOSE it will still be listed, whereas for VERIFY ..., REPAIR=YES it is considered to be a file in need of repair.
- ISAM: The file is unlocked only. If it is identified as open, the last-page pointer is set to the last PAM page written. If the file is on a mirror disk (see the "DRV" manual), equivalence between block contents is restored if required.  
 The file is closed.



=CHECK For NK-ISAM files only. Only files identified as open are considered. The file is unlocked, and the last-page pointer is set to the last PAM page written. The file is subsequently closed. Files with multi-functional blocks are checked for consistency, i.e. to determine whether writing of a multi-functional block has terminated abnormally. The defined secondary keys are checked to ensure that they are either complete or fully deleted. If the file is on a mirror disk (see the "DRV" manual), equivalence between block contents is restored if required.  
If no error is detected, the file is closed.  
For detailed information about NK-ISAM files please refer to the manual "DMS Introductory Guide and Command Interface" [8].

SUPPORT Selects files based on the type of volume on which they are located.

=PUBLIC Files on public volumes are to be selected for reconstruction.

=PRDISC Restricts reconstruction to those files which are on private volumes. The SUPPORT operand is only meaningful if a partially qualified file name or the name of a file generation group is specified for "filename1"

## WAIT    **Initiate conditional wait state**

Application group: Job variable functions (page 36)

The WAIT command is only available with the JV software product (see the "Job Variables" manual [11]).

### **Command description**

A job issuing a WAIT command is placed in the wait state until:

- either a particular condition is fulfilled, or
- the period specified in the TIME operand has elapsed.

The WAIT command is permitted both in interactive mode and in batch mode. The premature termination of a wait state can only be achieved by cancelling the job itself (CANCEL command).

The following applies when the event "condition fulfilled" occurs:

- In procedure and in batch mode, the next command following the WAIT command is executed.
- In interactive mode, a new command may be issued.

The following applies when the event "timeout" occurs:

- In procedure and batch modes, processing continues with the command whose label was specified with TIMELAB; otherwise a SPIN-OFF is triggered. In the event of a CALL procedure, the branch destination must be in the same procedure as the WAIT command. In batch mode, the branch destination must not be in a (DO/CALL) procedure if the WAIT command is part of an ENTER file. In both cases, the job is aborted when the branch destination cannot be found. The branch destination is not recognized if it is part of an ON/TIMEOUT command sequence.
- In interactive mode, a new command may be issued; the TIMELAB operand is ignored.

When a job has been placed in the wait state by means of the WAIT command, the wait state can be temporarily cancelled for the execution of ON statement sequences. The job is subsequently reset to the wait state.

If the command cannot be executed (due to an error during the syntax check or during the first evaluation of the conditional expression), a SPIN-OFF is triggered.

## Format and operand description

Operation	Operands
WAIT	[cond.exp] [,TIME=time] [,TIMELAB=.name]

- cond.exp** See page 358, "Conditional Expressions".  
"cond.exp" must not contain any special job variables. If it is omitted altogether, the job is placed in the wait state until the specified period has elapsed.
- TIME=time** Time interval in seconds;  $1 \leq \text{time} \leq 32767$  seconds. Default value: 600 seconds;  
Average accuracy: 200 msec.
- TIMELAB=.name** Specifies the branch destination in the event of a "timeout". ".name" must not be identical with the label of the WAIT command.

For examples, refer to the "Job Variables" manual [11].

## WHEN Enter conditional wait state

Application groups:

Job control (page 22)

Device and volume reservation (page 29)

### Command description

The WHEN command can be used to halt a job as a function of a condition.

The condition is set with the aid of user switches. The user switches for your own ID or a foreign user ID can be interrogated. Job processing resumes if the condition is met, or at the latest after the specified wait period.

The command is rejected in interactive mode.

The user switches are stored in the JOIN file. However, only the user switches from the JOIN file on the home pubset can be used.

The WHEN command releases all the resources of the job; device requests (SECURE and FILE commands) must be re-entered after the WHEN command.

#### *Exception*

Entries in the task file table (TFT) are not released. These entries, and the associated tapes and tape devices, must be released explicitly with the RELEASE command.

The following three situations may arise:

- a) A positive result is obtained from the comparison of the entries in the WHEN command and the current switch settings as soon as the command is processed, or neither the ON nor the OFF operand has been specified. The job then immediately continues processing with its next command.
- b) The first comparison produces a negative result but the requisite switches are set while the job is waiting. As soon as the system has ascertained this, the job continues processing with its next command.
- c) The result of the comparison is negative.  
If no TIME operand is specified in the WHEN command, the job waits until the end of the session.

If a TIME operand is specified, the wait time is restricted to this time limit. After this, the job continues processing at the next STEP command. If there is no next STEP command, the job is terminated.

If a time limit is specified by the TIME operand in the WHEN command and the job is on wait after an unsuccessful comparison, the actual wait time can be up to ten minutes longer because the relevant queue is only checked once every ten minutes by the system.

The WHEN command is rejected if

- a) the job has opened files on private volumes;
- b) a program is loaded.

If the WHEN command contains only the TIME operand, the command is simply ignored, and the job continues from the next command.

If the WHEN command contains only the "userid" entry, the command is simply ignored, and the job continues from the next command.

### Format and operand description

Operation	Operands
WHEN	<pre>[userid]  [ON={no       (no, ...)}]  [, OFF={no          (no, ...)}]  [, TIME=min]</pre>
userid	<p>User ID. Must be specified when checking the switches of another user.</p>
no	<p>Number of the user switch whose setting is to be checked. The 32 user switches are designated by decimal numbers from 0 to 31.</p>
ON=no (no,...)	<p>Specifies which switches are to be tested for an ON setting.</p>
OFF=no (no,...)	<p>Specifies which switches are to be tested for an OFF setting.</p>
TIME=min	<p>Waiting period (in minutes). The job run will resume at the latest after this period has elapsed. Maximum value: 9,999,999 minutes.</p>

**Example**

Runtime log for an ENTER job:

```
(IN)      LOGON RMA,ABTEIL05
...
(IN)      WHEN ON=20,TIME=1 _____ (01)
(OUT)    % EXC0864 JOB HAS LEFT ALL RESOURCES
(IN)     STEP
(IN)     FSTAT S.
(OUT)    0000003 :V:$PA123456.S.SAL.FILE1
(OUT)    :V: PUBLIC: 1 FILE. RES= 3, FREE= 2, REL= 0 PAGES
(IN)     LOGOFF NOSPOOL
(OUT)    % EXC0419 LOGOFF AT 1550 ON 90-09-04, FOR TSN 8078
(OUT)    % EXC0421 USED CPU TIME : 0.2499, SERVICE UNITS: 000002288
```

(01) If user switch 20 for user ID "RMA" is not switched to ON, the job releases its resources (EXC0864) and waits for that event in the WHEN queue for up to 1 minute.

If switch 20 remains in the OFF state during that period, a branch is made to the next STEP command, after which the FSTAT command is executed.

For a further example, see the SETUS command.

## WRITE-ACCOUNTING-RECORD      Write user accounting record

Application group: Job control (page 22 ff.)

### Command description

The WRITE-ACCOUNTING-RECORD command enables the user to write a user accounting record to the accounting file.

The accounting record may be:

- a UDAT accounting record with record extension, or
- a UACC accounting record with record identifier.

In order to evaluate the accounting records, you must use the appropriate programs ("System Administration" manual [1]). The system administrator can restrict the number of user accounting records for each task on a user-specific basis (MAXAREC parameter in the JOIN entry). The restriction applies to the entire command mode (except for program runs) for a task.

Default value: MAXAREC = 100; no more than 100 accounting records in the command mode for a task .

The "System Administration" manual [1] contains a detailed description of the accounting records.

**Format and operand description**

Operation	Operands
$\left\{ \begin{array}{l} \text{WRITE-ACCOUNTING-RECORD} \\ \text{WRITE-ACC} \end{array} \right\}$	$\left. \begin{array}{l} \text{RECORD-TYPE} = \left\{ \begin{array}{l} \text{USER-DATA (TEXT=C'text')} \\ \text{USER-ACCOUNTING-STEP (INFORMATION} = \left\{ \begin{array}{l} \text{C'recid'} \\ \text{X'recid'} \end{array} \right\}) \end{array} \right\} \end{array} \right\}$

**RECORD-TYPE** Specifies the type of accounting record.

**INFORMATION**={ ... }

recid = character string

**C:** Specified as character constant;

1 ≤ Length of 'recid' ≤ 8 characters.

**X:** Specified as hexadecimal constant;

1 ≤ Length of 'recid' ≤ 16 characters.

**TEXT**=C'text'

Character string whose length is 1 ≤ text ≤ 254.

The character "&" must not be used in the character string.

**USER-DATA**

A UDAT accounting record is written. The specified character string is entered in the accounting record as a record extension.

**USER-ACCOUNTING-STEP**

A UACC accounting record is written. The specified character string is entered in the accounting record as a record identifier.



## BS2000 system files

The (standard) file names SYSDTA, SYSCMD, SYSIPT, SYSLST, SYSLST01, SYSLST02, ..., SYSLST99, SYSOPT and SYSOUT denote (system) files used by the operating system for entering data or commands into the operating system or for outputting data. Each of these files is created by the task, and its initial (primary) assignment is to predefined input or output areas.

You as user can cancel the primary assignment and assign your own (cataloged) files to the (standard) file names. Some of the standard names can also be equated (see the SYSDTA command).

The following system files are available to a task for input purposes:

- SYSCMD** The commands for controlling job execution are read from SYSCMD. With primary assignment, SYSCMD is the data display terminal, in interactive mode; the ENTER or SPOOLIN file, in batch mode.
- SYSDTA** User program data, data or statements for a utility routine are read from SYSDTA. Compilers also use SYSDTA (depending on which operands are specified) as an input source for source programs.
- With primary assignment, SYSDTA is the data display terminal, in interactive mode; the ENTER or SPOOLIN file, in batch mode.
- SYSIPT** This system file is present for reasons of compatibility and corresponds to the system file SYSDTA.

The following system files are available to a task for output:

System files for output are created by the operating system as required and stored under your user ID. These consist of SAM files with the file names:

- S.OUT.tsn.mm.dd.yycon.hhmmss (SYSOUT)  
S.LST.tsn.mm.dd.yycon.hhmmss (SYSLST)  
S.OPT.tsn.mm.dd.yycon.hhmmss (SYSOPT)

where:

- tsn = TSN of the job
- mm.dd.yycon = Date (mm=month, dd=day, yy=year, con=consecutive day (calculated starting at the beginning of the new year).
- hhmmss = Time (hh=hour, mm=minutes, ss=seconds)

The memory used does not count as part of the allocated pubspace contingent. Files are automatically printed when the job is ended, and then deleted. You cannot access these files. The contents of the specified system file are (logically) deleted using the /ERASE \*SYSxyz (xyz=LST/OUT/OPT) command, while the catalog entry is retained. Any dummy system file is not printed.

**SYSOUT** All records output by means of the WROUT macro are written to SYSOUT. In the view of the operating system, these include, for example, all logging messages and error messages, and in batch mode, also the task log. Output (primary assignment) is to the S.OUT.tsn.mm.dd.yycon.hhmmss file in batch mode (file is printed at the end of the job); in interactive mode, files are also displayed at the terminal.

**SYSLST** All records output by means of the WRLST macro are written to system file SYSLST. In the view of the operating system, these are, for example, memory dumps and listings generated by language processors. In addition, all records to SYSOUT are also written to system file SYSLST, if the appropriate operands were specified in the LOGON or OPTION command. Output (primary assignment) is to the S.LST.tsn.mm.dd.yycon.hhmmss file. The file is printed at end of job.

**SYSLST01** System files SYSLST01,...,SYSLST99 provide the user with an  
**SYSLST02** additional means of outputting records using the WRLST macro.  
: Unlike system file SYSLST, these files are not created as  
: S.LST.tsn.... SAM files by the operating system. They are only  
: effective if the user has assigned cataloged files to them. Their  
**SYSLST99** primary assignment is to (system) file SYSLST.

**SYSOPT** All records output by means of the WROUT macro are written to SYSOPT. Output (primary assignment) is to the S.OPT.tsn.mm.dd.yycon.hhmmss file. The file is output to floppy disk when the job is ended.

System files scheduled for output can also be output beforehand (START-SPOOL operand in the PRINT or PUNCH command).

In the PRINT, PUNCH or ERASE command, or in the corresponding macros, the (standard) file names SYSOUT, SYSLST and SYSOPT can also be specified when cataloged files are assigned to them.

Fig. SY-1 BS2000 system files

### Access to system files

System files can be used by user programs for inputting and outputting data, except for the system file SYSCMD, which can only be accessed by the system itself via privileged macros. The following table lists the macros or statements required to access system files:

System file	Macros accessing system files	Language statements referring to system files:				
		Assembler	COBOL	FORTRAN	ALGOL	PL/I
SYSCMD	privi- leged	-	-	-	-	-
SYSDTA	RDATA	ACCEPT data-name {FROM SYSIN {FROM TERMINAL	READ (1,... READ (5,... READ (97,...	All in- put pro- cedures with DSN 3 (except GET)	GET without FILE option	(DEVICE=) no explicit statement
SYSIPT	RDCRD	ACCEPT data-name; ACCEPT data-name FROM SYSIPT		All in- put pro- cedures with DSN 0 (except GET)	-	-
SYSOUT	WROUT	DISPLAY data-name {UPON TERMINAL {UPON SYSOUT	WRITE (2,...	All out- put pro- cedures with DSN 4 (except PUT)	DISPLAY data- name (no REPLY)	DSPLY

System file	Macros accessing system files	Language statements referring to system files:				
	Assembler	COBOL	FORTRAN	ALGOL	PL/I	RPG
SYSLST	WRLST	DISPLAY data-name; DISPLAY data-name UPON SYSLST	WRITE (6,... WRITE (99,...	All out-out procedures with DSN 1 (except PUT)	PUT without FILE option	(DEVICE=) PRINTER, no explicit statement
SYSOPT	WRTOT	DISPLAY file-name {UPON SYSPUNCH; {UPON SYSOPT	WRITE (7,... WRITE (98,...	All out-put procedures with DSN 2 (except PUT)	-	-

### Primary assignment and readdressing of system files

Usually, system files are given a predefined assignment. This primary assignment can be changed by means of the commands summarized in the following table. Examples can be found in the descriptions of the individual commands.

System file	System file assignments Primary assignment	Further assignments	Commands to change assignments
SYSCMD	Interactive: terminal; batch: spoolin file "S.INTsn" (spooled in via magnetic tape device, disk device, batch terminal or ENTER file)	Cataloged disk file (SAM/ISAM)	DO command: changes assignment to cataloged file. CALL command: changes assignment to cataloged file. ENDP command (procedure files only): returns file to primary assignment following DO command or to last procedure step left via CALL command.
SYSDTA	See SYSCMD primary assignment	Cataloged disk file (SAM/ISAM) or card reader	SYSFILE command: assigns file to cataloged file, to SYSCMD or back to primary assignment.  ENDP command: returns file to assignment valid before the procedure level was called.
SYSIPT	Interactive: only set up if required. Batch: See SYSCMD primary assignment.	Cataloged disk file (SAM/ISAM)	Same as SYSDTA.
SYSOUT	Interactive: terminal; Batch: temporary system file S.OUT. ..., which is printed out at end of job and then deleted.	Batch: cataloged file which is not printed out automatically (PRINT command required).	As with SYSLST, but only in batch mode.

System file	System file assignments Primary assignment	Further assignments	Commands to change assignments
SYSLST	Temporary system file S.LST. ..., which is printed out at end of job and then deleted (only set up if required).	Cataloged file which is not automatically printed out (PRINT command required).	SYSFILE command: assignment to a cataloged file or return to primary assignment. ENDP command: returns file to assignment valid before the procedure was called.
SYSLST01 . . . SYSLST99	System files. Primary assignment = assignment of SYSLST.	See SYSLST; assignment to each other also possible.	Same as SYSLST.
SYSOPT	Temporary system file S.OPT. ..., which is printed out at end of job and then deleted (only set up when required).	Cataloged file which is not automatically printed out (PUNCH command required).	Same as SYSLST.





# Appendix

## Device type code

- 1. = FAMILY code
- 2. = Device channel class
- 3. = Device type code

Device family	FAMILY name	1.	2.	3.	Device type	Device designation/ Product number
Operator terminals	CONSOLE	00	S	02	CON3027	BST 3027-1, -2 BST 3027-101, -102
				03	CON3027C	BST 3027-11, -21 BST 3027-111, -121 BST 3027-LRC
			I	04	CON04	3027 terminal emulation for CPU with bus periphery
				0A	CON38	3809/3886 75407-3, -4, -5
				0B	CON3803	75407-1, 3886-2, -3 (Hardcopy printer at the SVP)
				0C	CON3888	Hardcopy 3888-3 (for 3886 subconsole) at the 3803-90 cluster controller 75407-1

# Appendix

Device family	FAMILY name	1.	2.	3.	Device type	Device designation/ Product number
Printers	PRINTER	20	S	24	PRPND	3350-1 3352-1
				26	PRLS333	3337-51, 3338-51, -511, -512, -521, -522 3339-51, -512, -52, -522
				27	PRPSHP	3351-21, -211 3353-21, -211
				2C	PRL3365	3365-11
			I	28	PRPIXH	2090-2, 2140-2
				29	PRL29	Bus printer for CPU with bus periphery
				2A	PRLI333	3338-531, -53, -532 3339-53, -532
				2B	PRPIHP	3351-23, -231 3353-23, -231
				2C	PRL3365	3365-12
				2F	PRL3348	3348-120, 3349-120
Special devices	FAM50	50	S/I	51	DSVP1	SVP hard disk
			I	52	DSVP2	SVP hard disk on C40
			S	53	TD8170	8170-21 (MSN)

Device family	FAMILY name	1.	2.	3.	Device type	Device designation/ Product number		
Tele-processing	TD	60	S	61	TD960	9631-1,-2,-3		
				62	ZAS-DUMP	9631-50,-51,-52,-55		
				63	ZAS-BCAM			
				6C	ZAS-SIN	TRANSDATA CPU port controller with SINIX connection		
				6D	ZAS-LAN	9632-100		
				6E	DAST	3612		
			I	61	TD960	9631-1,-2,-3		
				62	ZAS-DUMP	9631-60,-61,-62,-65		
				63	ZAS-BCAM			
				64	SKP			
				6D	ZAS-LAN	9632-200		
				6E	DAST	3801-B		
			Physically supported devices				71 . . . . 7F	"name of foreign device"  These names are defined by means of ADAM. Assignment of names to device type codes by means of UGEN statement ADT.
			Floppy disk devices	DISKETTE	90	S	92	FD30243
93	FD3171	3171 with 31712 1) opt <sup>i</sup> o <sup>n</sup>						
I	9B	FD75407				75407-2 (C40)		

1)

A CTL and two DVC statements (with ascending addresses) must be specified for these devices at system generation time.

Device family	FAMILY name	1.	2.	3.	Device type	Device designation/ Product number
Disk devices <sup>1)</sup>	DISK	80/ A0				
		80	I	8F	D3475-8F	74305-12, -13, -140, -141, -150, -151 (C30)
		A0	S/I	A1	D3439-10	3439-10, -12
				A2	D3436	3436, 3436-2, -10, -12
				A3	D3437	3437, 3437-2
			I	A5	D3435	3435 (C40)
				A7	D3490-10	3490-1A4, -1A8, -1B4, -1B8, -1BC
			S/I	AB	D3475	3475-1, -2, -3
			I	AC	D3480	3410 <sup>2)</sup> (high-speed memory)
			S/I	AC	D3480	3480-1, -2, -11, -12, -111, -112 3848-A4, -B4, -AD4, -BD4
				AD	D348E	3480-21, -22 3848-AE4, -BE4
			I	AE	D348F	3480-131, -132
		AF		D3490-20	3490-2A4, -2A8, -2B4, -2B8, -2BC	

1)

One DVC statement per disk drive must be specified for disk devices at system generation time.

2)

The DYNREC=NO operand must be specified for SSD 3410 in the CTL statement

Device family	FAMILY name	1.	2.	3.	Device type	Device designation/ Product number
Tape devices	TAPE	B0/ C0/ E0				Controller + drive Unit, + Element +
unimodal tape devices	UNMTAPE	B0	S	B2	UM1600	3570 + 3530 3571 + 3531
				B4	UM6250	3513 <sup>1)</sup> + 3557, 3559
			I	B4	UM6250	3514 <sup>1)</sup> + 3557, 3559
				B7	UM1600-1	3534
			I	B9	UMVID-1	MTC 2.1 Gbyte video 8
				BA	UMSC-1	MTC 155 Mbyte (only for SIR and ARCHIVE)
Magnetic tape cartridge devices	MBK	C0	I	C1	3580	3580-A10 + 3580-B10 3580-A20 + 3580-B20 3590-D31 3590-D32
				C2	3590	3580-A10 <sup>2)</sup> + 3580-B10 3580-A20 <sup>2)</sup> + 3580-B20 3590-D31 <sup>3)</sup> 3590-D32 <sup>3)</sup> 3590-A01 + 3590-B02/-B04 3590-A02 + 3590-B04/-B04
				C4	3590E	3590-D41 3590-D42 3590-A10 + 3590-B20/-B40 3590-A20 + 3590-B20/-B40

1)

MBS must be specified for the type operand in the CTL statement for these devices.

2)

With 35830 option for improved recording mode (VAV)

3)

With 35930 option for improved recording mode (VAV)

Device family	FAMILY name	1.	2.	3.	Device type	Device designation/ Product number
Tape devices	TAPE	B0/ C0/ E0				Controller + drive Unit, + Element +
Bimodal tape devices	BIMTAPE	E0	S	E2	BM1662	3513 <sup>1)</sup> + 3557, 3559
			I	E2	BM1662	3514 <sup>1)</sup> + 3557, 3559
			S	E2	BM1662	3515 + 3525 3516 + 3526 3517-1 + 3527-1 3519 + 3529 3535 + 3525 3536 + 3526 3537-1 + 3527-1
			I	E2	BM1662	3517-3 + 3527-3 3519-3 + 3529
			S	E3	BM1662S	3518 + 3528 3538 + 3528
			I	E4	BM1662S1	3506 (C40)
				E8	BM1662FS	3504-625

1)

MBS must be specified for the type operand in the CTL statement as the value for these devices.

*device channel classes*

- S: Block multiplexer channel type 1 (SBL) or  
Byte multiplexer channel type 1 (SBY)
- I: Block multiplexer channel type 2 (IBL) or  
Byte multiplexer channel type 2 (IBY) or  
Emulation of multiplexer channel type 2 (bus channel)

Disk devices are always connected to block multiplexer channels.  
Magnetic tape devices can be connected to block multiplexer channels and byte multiplexer channels.

Volume type	Meaning
T1600	Tapes with recording density of 1600 BpI (device type codes: B2, E2, E3, E4, E8)
T6250	Tapes with recording density of 6250 BpI (device type codes: B4, E2, E3, E4, E8)
WORK TAPE }	Tapes with recording density of 1600 or 6250 BpI
TAPE-C1	Magnetic tape cartridge, 18-track (device type codes: C1, C2)
TAPE-C2	Magnetic tape cartridge, 18-track, compressed (device type code: C2)
TAPE-C3	Magnetic tape cartridge, 36-track (device type code: C4)
TAPE-C4	Magnetic tape cartridge 36-track, compressed (device type code: C4)
TAPE-V1	Magnetic tape cartridge 2.1 Gbytes video 8 (device type code: B9)
TAPE-CS1	Magnetic tape cartridge 155 Mbytes (device type code: BA)

## Meaning of output columns for the SHOW commands

Meaning of the output columns for the SHOW device management commands:

Keyword	Meaning
ACTION	<p>Indicates which (re)allocation operation is running for a volume from the volume monitor, activated by:</p> <ul style="list-style-type: none"> <li>- operator intervention (accidental mounting of a used volume)</li> <li>- commands (DETACH-DEVICE, MOVE-DISK,...)</li> <li>- Device Error Recovery (DER), e.g. with INOP</li> <li>- user request (MOUNT message for unmounted volumes)</li> </ul> <p>The following states may occur:</p> <p>CANCELLED: use of a tape or disk is permanently locked; the interrupt will not be retracted</p> <p>DISMOUNT: a REMOUNT message for the same volume is pending on another device, or a REMOUNT or MOUNT message for another volume is pending for the same device</p> <p>INOP: the device is not available (inoperable)</p> <p>MOUNT: the response to a MOUNT message is still pending for the volume</p> <p>NO ACTION: no interrupt</p> <p>NO DEVICE: no device allocation exists for a volume due to a preceding reconfiguration command (DETACH-DEVICE, REMOVE-DEVICE-CONNECTION)</p> <p>POSITION: a tape currently in use is being repositioned</p> <p>PREMOUNT: the response to a PREMOUNT message for the volume in question is still pending</p> <p>RECOVER: an unspecified interrupt handling operation is taking place for the volume currently in use</p> <p>REMOUNT: remounting is taking place for the volume</p> <p>SNATCHED: the allocation was passed to another task; the allocation was retracted by the device owner</p> <p>SVL-UPDATE: the system occupancy log is currently being stored on the disk</p> <p>UNLOCK: an UNLOCK job is being executed to extract a system ID stored in the SVL</p> <p>WP-MISSING: the write-enable ring must be mounted (tape) or the write lock must be cancelled (disk)</p>



Keyword	Meaning
ACCESS	<p>PPD: determines the use of the disk in PPD mode (PPD: Protected Private Disk; chargeable software product)</p> <p>WRITE: the disk can be read- and write-accessed</p> <p>ALL: not until the disk has been allocated is the final ACCESS value determined, depending on the generation characteristic of the device on which the private disk is mounted:</p> <p>POOL=NO SH sets ACCESS=WRITE</p> <p>POOL=SW sets ACCESS=PPD</p> <p>This value is not dependent on the setting of the "WRITE INHIBIT" switch.</p>
ADMISSION-TIME	Date and time at which the task was entered in the SECURE queue (yyyy-mm-dd hh:mm:ss)
ASS-TIME/ ASSIGN-TIME	<p>SH-DISK (INF=PAR) defines the time at which a private disk with mode USE=DMS will be allocated or deallocated</p> <p>SH-DISK-DEF defines the time of disk allocation or deallocation for all disks which do not have an explicit setting for this value</p> <p>USER: allocation requests or returns made by the user</p> <p>OPERATOR: if the disk is online and not yet allocated by the system, allocation is activated immediately</p>
ATT	Number of devices in the "ATTACHED" state which belong to the device type specified in the output (regardless of the allocation)
AVAIL	Number of available devices of the device type specified in the output

Keyword	Meaning
CONF-STATE	Configuration state of the specified device, indicating whether or not it is available  ATTACHED: the device can be used by the system for input/output DETACHED: the device cannot be used by the system for input/output DET-PENDING: the device will be declared detached for the system when it is no longer being used INVALID: the device cannot be reached via any path
CTL-CHN-IOP	Inner virtual connection of the device to the I/O processor via controller (CTL) and channel (CHN)
DET	Number of generated devices of the specified type which are not available due to a configuration state "DETACHED"
DET-P	Number of devices of the relevant type which are still required for user requests and which will be DETACHED following deallocation; reallocation of these devices is no longer possible
DEV-A	Type of device allocation: FREE: the device has not yet been allocated, and is freely available DMS: the relevant device is implicitly allocated by a DMS allocation on the private disk which is mounted on it PUBLIC: the device is implicitly allocated by a public disk which is mounted on it tsn: TSN of the job to which the device has been exclusively allocated; it was requested by the SECURE-RESOURCE-ALLOCATION command (UNIT=operand), or, in the case of disk storage devices, the occupying job is using the allocated disk for a USE-SPECIAL application (PHASE=IN-USE or MOUNT)
DEV-TYP[E]	Device code (see device table)
DVC	Mnemonic device name of a specified hardware unit.
ICUU	Path address (IOP, CHN, CTL, DVC) to a device.

Keyword	Meaning
INNER CONNECTION	Describes the availability of the connection generated from the specified unit to all inner units directly connected to it (seen from the CPU/IOP direction). The following states may occur: INCLUDED: the connection (path) is available to the system for I/O operations. REMOVED: the path is not available to the system for I/O operations. REM-PENDING: the path will be declared unavailable when no longer used by the system.
I/O-PATH-STATE	Availability of a complete I/O path (from IOP via CHN, CTL to the device). AVAILABLE: path is available NOT AVAILABLE: path is not available
LABEL	Type of volume label  STD: volume with standard labels BS1000: disk with BS1000 labels TAPE-MARK: tape starts with a tape mark NON-STD: the label does not have any of the previously mentioned attributes
MNEM	Shows the mnemonic device name defined at generation time
NAME/ID	Provides information on the job name of the addressed job, or on the user ID under which the job is running
OP-CTL/ OPERATOR-CONTROL	Indicates whether the operator wants to be kept informed of initial disk allocations of tasks (with the option of rejecting these allocation requests): SHARE, EXCL, ALL, NO
OUTER CONNECTIONS	Describes the configuration state of the connections generated from the specified unit to all external units directly connected to it (seen from the end-device direction). INCLUDED: the connection (path) is available to the system for I/O operations REMOVED: the path is not available to the system for I/O operations REM-Pending: the path will be declared unavailable when no longer used by the system

Keyword	Meaning
PHASE	<p>Provides information on the tape and disk monitoring modes.</p> <p>ONLINE: the volume is mounted but not allocated</p> <p>PREMOUNT: the volume is allocation; a device allocation exists for it for previous or future use</p> <p>MOUNT: the volume is already allocated but still has to be made available by the operator</p> <p>IN-USE: the volume is deallocated for use (exception: ACTION=CANCELLED)</p> <p>Volume monitoring takes place for the allocation states IN-USE and PREMOUNT (a volume will always be monitored if a valid allocation exists for it). The tasks of volume monitoring are as follows:</p> <ul style="list-style-type: none"> <li>- it guarantees a device allocation for tapes in the PREMOUNT state;</li> <li>- it requests the operator to remount a volume which is considered allocated but is currently not accessible (INOP);</li> <li>- it takes care of cancellation of the NO-DEVICE state as soon as a device of the requisite type becomes available;</li> <li>- it initiates automatic repositioning of tapes if the operator has made a mistake (e.g. unloaded the wrong tape device)</li> </ul>
POOL	<p>Defines the availability of a device in relation to two or more systems</p> <p>NO: the device is only available from the home system; it is not possible to access a volume mounted there from another system</p> <p>SH: the device is generated as a rule for two or more systems (device with multiprocessor interface); any disk mounted on a device with this generation characteristic is operated by default as an SPD disk</p> <p>SW: this device is generated as a rule for two or more systems;</p> <p>with disk storage devices:</p> <ul style="list-style-type: none"> <li>- device with multiprocessor interface: private disks mounted on it are operated system-exclusively by default (non-SPD mode);</li> <li>- with PPD mode: only read access is permitted</li> </ul> <p>with other devices:</p> <ul style="list-style-type: none"> <li>- parallel operation is impossible or impractical due to lack of hardware support (tape devices). For devices with this generation characteristic, the operators of the participating systems must ensure that they are ATTACHED in one system only.</li> </ul>

Keyword	Meaning
PRE-/IN-USE	Number of devices of the defined type which are allocated implicitly by volumes of the corresponding phase (PREMOUNT, MOUNT(ing), IN-USE)
RES-BY-MN	Number of devices of the relevant type which were reserved by a user with the command SEC-RES UNIT=mn
RES-BY-TYPE	Indicates how many free devices of the specified type are required to deal with reservation and allocation requests which have already been granted
RESOURCES REQUESTED/ COLLECTED	List of devices or volumes specified with /SEC or list of devices or volumes already reserved by the collector task
SVL-ALLOC	Actual allocation mode of the disks (system-shareable or system-exclusive). This need not be identical to the entry made by the operator using the command SET-DISK VOL=vsn,SYS=...
SYSTEMS	System IDs of the systems occupying the disk. The allocation is stored on the SVL of the disk.
SYS-ALLOC	<p>Determines the mode in which a private disk with USE=DMS is to be used by the home system with reference to other systems.</p> <p>Permissible operating modes:</p> <p>EXCLUSIVE: other systems are excluded from using the disk</p> <p>SHAREABLE: other system may access the disk (SPD mode); synchronization with other systems takes place with regard to space and file utilization.</p> <p>Catalog locks appear in the F1 label of the disk</p> <p>ALL: the system allocation mode is derived from the generation characteristic of the device</p>

Keyword	Meaning
TASKS-WITH-RESERVATIONS	List of tasks with the number of devices of the specified type which have been reserved or allocated
TIME-STAMP	Date and time at which the SVL of the disk last detected an initial allocation (date/time of system ID most recently stored in the SVL). This time punch is used together with the VSN to identify a disk.
TIME-WEIGHT	Provides information on the wait period which was set by /START-RES and which has an effect on the calculation of the urgency (weight) for the collector task
TSK-PRIO	Priority of the job in question
TSK-TYPE	Provides information on whether a batch, interactive or RFA-created task is involved on the remote processor (SECURE requests were sent from a different processor)
TSN	Task sequence number
TYPE	<p>Describes the device type of the volume on which information is being requested; the device type can be influenced, not only by a request from you (SECURE, FILE, ...), but also by the following events:</p> <p>online event: the activation interrupt allocates the volume to a device whose DEVICE-TYPE then determines the device type of the volume in a VSN request.</p> <p>SET-DISK command: the device type is predefined prior to volume allocation.</p>
UN-CLASS	<p>Device class to which the specified unit belongs:</p> <p>DVC: describes an end device (disk, tape, printer, etc.)</p> <p>CHN: describes a channel</p> <p>CTL: describes a controller</p> <p>IOP: describes an I/O processor</p> <p>CPU: describes a central processing unit</p>

Keyword	Meaning
UN-TYPE	Generic type for "DEVICE-TYPE". It includes not only the set of possible device types, but also the values of all CTL, CHN, IOP and CPU types.
USE	<p>Provides information on</p> <ul style="list-style-type: none"> <li>- the allocation mode of a mounted volume</li> <li>- the degree of monitoring</li> <li>- the scope of monitoring during allocation by the monitors</li> </ul> <p>Permissible values:</p> <p>DMS: The volume is occupied by one or more DMS applications. Only readable files are accepted for allocation, i.e. only standard label disks can be processed and tapes with or without STD labels can be processed provided they can be identified uniquely.</p> <p>By default, the task allocation mode for USE=DMS is task-shareable for private disks and task-exclusive for tapes. Each operator action during PHASE=IN-USE leads to a REMOUNT-RECOVER and causes tapes to be repositioned. The system ensures that only one volume with a given VSN is allocated in DMS mode.</p> <p>SPECIAL: The volume is occupied by a special application (privileged application, e.g. VOLIN, INIT, debugging routines, FDDRL, etc.). The task and system allocation mode is EXCLUSIVE. The special application can switch off checking operations in connection with the allocation (VOLIN, INIT) or monitoring functions such as repositioning or MOVE (online FDDRL organizes this itself). The VSN is not checked for uniqueness.</p> <p>WORK: The mounted tape is used as a work tape (it is made available to the DMS user for processing WORK files).</p>
USER-ALLOC/ USER- ALLOCATION	Indicates which allocation requests are permitted for a user (task-shareable, task-exclusive, ALL) for a private disk operated with USE=DMS.

Keyword	Meaning
VOL-A	<p>Provides information on attributes of public disks or allocation modes of private disks</p> <p>for public disks:</p> <p>PAGING: The disk is part of the occupied pubset and is used for paging purposes.</p> <p>PUBLIC: The disk is part of the occupied pubset.</p> <p>SYSRES: Public disk of the home pubset on which the TSOSCAT catalog begins.</p> <p>for private volumes:</p> <p>FREE: No user is accessing the volumes at the moment.</p> <p>EXCL: The private volume is allocated exclusively to a user job.</p> <p>SHARE: The private disk is occupied by one or more jobs; additional requests are permitted.</p>
VSN	<p>Volume serial number, the "name" of a volume; it is defined when a volume is initialized (VOLIN, INIT). If the volume does not have a readable label, or if no VSN was given in the volume request, synonyms may be output.</p> <p>Permissible values:</p> <p>&lt;vsn&gt;: the VSN of a volume as defined in VOLIN or INIT</p> <p>UNKNO: the volume does not have a standard BS2000 label</p> <p>SCRAT: the volume request was not given a VSN (e.g. FILE command without VOLUME operand for tapes)</p> <p>WORK: the tape in question was requested with /FILE.....,DEVICE=WORK</p>



Keyword	Meaning
VTOC-LOCK	<p>Software lock on a private disk which prevents the following actions:</p> <ul style="list-style-type: none"><li>- Allocation or deallocation of the disk by a system (entry or removal of a system ID in the SVL)</li><li>- Access operations to the F1 label (directory of files initialized on the disk)</li><li>- Access operations to the F5 label (overview of occupied and free pages on the private disk).</li></ul>
VTOC-SYS	<p>System ID of the system which currently has the VTOC lock for the disk and thus temporarily excludes other systems from space and catalog operations on this disk. The VTOC system is stored in the SVL of the disk.</p>
VTOC-TSN	<p>Job of the home system, causing the home system to allocate the VTOC lock</p>
WAIT-TIME	<p>Period of time that the specified task has been waiting for deallocation of the requested resources (hh:mm:ss)</p>



# Abbreviations

AID	Advanced Interactive Debugger
AFR	Access From Remote
AVR	Automatic Volume Recognition
BTAM	Basic Tape Access Method
CJC	Conditional Job Control
CPU	Central Processing Unit
CLT	Communication Link Table
CVT	Current Volume Table
DCAM	Data Communication Access Method
DCM	Data Communication Methods
DLL	Dynamic Linking Loader
DMS	Data Management System
EAM	Evanescent Access Method
EBCDIC	Extended Binary Coded Decimal Interchange Code
ECR	Executive Communication Region
EITC	Extended Inter-Task Communication
ELDE	static loader
EOF	End of File
ES	Event Switch
ESA	Executive Storage Area
ETX	End of TeXt
FCB	File Control Block
FCP	File Control Processor
FGG	File Generation Group
FIFO	First In - First Out
FT	File Transfer
HSMS	Hierarchical Storage Management System
IDA	Interactive Debugging Aid
II	Information Indicator
ISAM	Indexed Sequential Access Method
ISD	Internal Symbol Dictionary
ITC	Inter-Task Communication
ITN	Internal Task Number
JV	Job Variable
LIFO	Last In - First Out
LMR	Library Maintenance Routine

## Abbreviations

---

Mb	Megabyte: 1 million bytes
MC	MRS Communication
MCLP	Macro Command Language Processor
MPVS	Multiple Public Volume Set
MSCF	Multiple System Control Facility
OPR	Overwrite PRotection
PAM	Primary Access Method
PC	Program Counter
PCB	Process Control Block
PVS	Public Volume Set
PT	Program Table
RFA	Remote File Access
RS	Return Switch
RTIO	Remote Terminal Input Output
SAM	Sequential Access Method
SI	Secondary Indicator
SVC	Supervisor Call
TCB	Task Control Block
TCS	TeleCommunication System
TFT	Task File Table
TOS	Tape Operating System
TSET	Tape SET
TSN	Task Sequence Number
TST	Tape Set Table
UPAM	User PAM
USASCII	USA Standard Code of Information Interchange
VFB	Vertical Format Buffer
VSN	Volume Serial Number
VTOC	Volume Table Of Contents

# References

- [ 1] BS2000  
**System Administrator Commands (ISP Format)**  
User Guide
- Target group*  
System administration.
- Contents*  
ISP commands for system administration.
- Applications*  
System administration.
- [ 2] BS2000  
**System Operator's Guide**  
User Guide
- Target group*  
BS2000 operators.
- Contents*  
Responsibilities of the operator for a BS2000 installation, including:
- system initialization and termination
  - operator commands
  - device management
  - dump routines.
- Applications*  
Computer center.
- [ 3] BS2000  
**Introductory Guide for System Users**  
User's Guide
- Target group*  
BS2000 users.
- Contents*  
Introduction to BS2000; description of the most frequent user commands; introduction to using the utility routines and software products EDT, SORT, ARCHIVE, TSOSLNK, LMS and PERCON; notes for the programmer.
- Applications*  
BS2000 interactive mode and batch mode.

- [ 4] BS2000  
**Interactive Debugging Aid (IDA)**  
Reference Manual
- Target group*  
Programmers.
- Contents*  
Description of the commands and macros for the Interactive Debugging Aid (IDA).
- Applications*  
BS2000 interactive mode.
- [ 5] BS2000  
**Executive Macros**  
User Guide
- Target group*  
BS2000 assembly language programmers (non-privileged); system administrators.
- Contents*  
All Executive macros in alphabetical order with detailed explanations and examples; selected macros for DMS and TIAM; macro overview according to application areas; comprehensive training section dealing with eventing, serialization, inter-task communication, contingencies.
- Applications*  
BS2000 application programs.
- [ 6] TRANSDATA  
**DCAM**  
**Program Interfaces**  
Reference Manual
- Target group*  
Managers, application planners, programmers, system and network administrators
- Contents*  
Description of the Data Communication Access Method DCAM

- [ 7]    **TRANSDATA**  
      **File Transfer in BS2000**  
      User Guide
- Target group*  
          Users and administrators of FT-BS2000, BS2000 system administrators
- Contents*  
          Capabilities and functions of FT-BS2000; description of the user and administrator command interfaces; description of the program interface; all the commands required for FT-BS2000 operation
- [ 8]    **BS2000**  
      **DMS Introductory Guide and Command Interface**  
      User Guide
- Target group*  
          Non-privileged BS2000 users.
- Contents*  
          Functions of DMS in BS2000; processing of disk and tape files; access methods UPAM, SAM, BTAM, EAM, ISAM; DMS commands.
- [ 9]    **BS2000**  
      **DMS Assembler Interface**  
      User Guide
- Target group*  
          Non-privileged BS2000 users/assembly-language programmers.
- Contents*  
          Functions of DMS in BS2000 (at macro level); processing of disk and tape files (at macro level); access methods UPAM, SAM, BTAM, EAM, ISAM (including action macros); file processing macros.
- [10]    **BS2000**  
      **Diagnostics Handbook**  
      User Guide
- Target group*  
          BS2000 programmers and system administrators.
- Contents*  
          Components and tools for identifying, logging and analyzing software error data, namely:
- dump generators (CDUMP, SLED, SNAPSHOT)
  - dump analyzers (DAMP, SODA)
  - logging tools (SERSLOG, AUDIT, trace manager)
  - log evaluator (ELFE).
- Applications*  
          Software diagnosis.

- [11] **BS2000 Job Variables**  
User Guide
- Target group*  
BS2000 users.
- Contents*  
Applications for job variables in controlling and monitoring jobs and program runs; conditional job control; all the necessary commands and macros; application examples.
- Applications*  
BS2000 timesharing mode.
- [12] **RFA (BS2000)**  
User Guide
- Target group*  
BS2000 RFA users (non-privileged).
- Contents*  
Functions of RFA (Remote File Access); commands.
- Applications*  
Remote file access, BS2000 interactive mode.
- [13] **BS2000 System Installation**  
User Guide
- Target group*  
BS2000 system administrators.
- Contents*  
New installation; version changeover; generation of a new public volume set; generation of a subsystem catalog; statements for SIR and UGEN.
- Applications*  
System administration, computer center.
- [14] **LMS (BS2000)**  
Reference Manual
- Target group*  
BS2000 users.
- Contents*
- Description of the LMS statements in ISP format for creating and managing PLAM libraries
  - Storage using the delta method
  - Invocation as a subroutine from COBOL and Assembler programs.
- Applications*  
Interactive/batch mode.



- [15] **MSCF (BS2000)**  
**Multiprocessor System**  
User Guide
- Target group*  
BS2000 system administrators, operators, end users.
- Contents*  
Generation, operation, management and application of MSCF for processors to be included in a computer network.
- Applications*  
System management, computer center, network participation.
- [16] BS2000  
**Utility Routines**  
User Guide
- Target group*  
BS2000 users (non-privileged).
- Contents*  
Utility routines for non-privileged BS2000 users.
- Applications*  
BS2000 timesharing mode.
- [17] TRANSDATA  
**DCAM**  
**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

- [18] **TRANSDATA**  
**9749, 9750, 9752 Data Display Terminals**  
**Programmer's Interface**  
User's Guide
- Target group*  
Programmers (applications programmers) who wish to program outputs to display terminals and interpret/analyze inputs from such terminals
- Contents*  
Notes on configuring TRANSDATA networks; description of the terminal functions; notes on the physical and logical programming of these functions; description of the message format; data exchange with printers; description of the software product PLUS
- [19] **BS2000**  
**SPOOL Part 1, System Description**  
User Guide
- Target group*  
SPOOL users.
- Contents*  
Description of printer output control, SPOOL commands, SPOOL macros, laser printer functions, remote batch processing.
- [20] **RSO (BS2000)**  
User Guide
- Target group*  
System administrators, device administrators, end users.
- Contents*  
Descriptions of
- the user and administrator commands for performing remote SPOOL jobs (without SPOOL parameter commands);
  - installation of the software product RSO;
  - generation of RSO printers;
  - the RSOSERVE utility routine;
  - operation of the 9025 Electronic Page Printer;
  - the RSO messages.

- [21] **HSMS (BS2000)**  
User Guide
- Target group*  
BS2000 users, BS2000 system administrators.
- Contents*  
Description of the data saving, archiving and migration functions; tasks of HSMS administration; description of the statements.
- [22] BS2000  
**Binder-Loader-Starter (BLS)**  
User Guide
- Target group*  
Software developers.
- Contents*  
The binder-loader-starter (BLS) system consists of the following functional units:
- linkage editor BINDER
  - dynamic binder loader DBL
  - static loader ELDE
- The various sections contain functional descriptions and examples, plus a reference section with statements, commands and, where applicable, macros.
- [23] **AID (BS2000)**  
**Advanced Interactive Debugger**  
**Debugging on Machine Code Level**  
User's Guide
- Target group*  
BS2000 programmers.
- Contents*  
Description of all the AID commands available for debugging on machine code level; messages.
- Applications*  
Debugging of programs in interactive/batch mode.

- [24] BS2000  
**System Messages**  
User Guide
- Target group*  
BS2000 users.
- Contents*  
Standard format messages of the BS2000 control system, including SPOOL, RSO, SDF; standard format messages of the software products DCAM, TIAM, RBAM.
- [25] BS2000  
**Computer Center Utility Routines**  
User Guide
- Target group*  
BS2000 system administrators.
- Contents*  
Utility routines available under the system administrator ID for the purpose of controlling and monitoring the operating system.
- Applications*  
System administration, computer center.
- [26] TRANSDATA  
**FTAC-BS2000**  
**Extended Access Control for File Transfer**  
User's Guide
- Target group*  
Users and administrators of FT-BS2000 and FTAC-BS2000, data protection managers
- Contents*  
Capabilities, functions and use of FTAC-BS2000; all the commands and procedures required when using FTAC-BS2000

- [27] BS2000  
**Systemübersicht** \*  
Technische Beschreibung  
(System Overview, Technical Description)

*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.

*Contents*

- Characteristics of BS2000 (application and performance features, interfaces, resources, internal structure and execution sequences).
- Possible hardware configurations.
- BS2000 subsystems (nucleus, Data Management System, job management system, programming system, Data Communication System, transaction monitor, system management, service system).

*Order number*

U3210-J-Z53-1

- [28] BS2000  
**Nucleus**  
Technical Description

*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.

*Contents*

- Hardware functions and HSI (Hardware-Software Interface).
- Functions and principles of implementation of
  - the hardware drivers
  - process management
  - address space and paging
  - the I/O system
  - error recovery and reconfiguration
  - PCS (Performance Control Subsystem).

*Order number*

U3211-J-Z53-1-7600

- [29] BS2000  
**Datenverwaltungssystem** \*  
Technische Beschreibung  
(Data Management System, Technical Description)

*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.

*Contents*

Functions and principles of implementation of

- the DMS services
- memory management
- file management
- the access methods
- data protection.

*Order number*

U3212-J-Z53-1

- [30] BS2000  
**Auftragsverwaltungssystem** \*  
Technische Beschreibung  
(Job Management System, Technical Description)

*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.

*Contents*

Functions and principles of implementation of

- the job management system
- the multiprocessor system
- the job variable system
- SPOOL
- the accounting system.

*Order number*

U3213-J-Z53-1

- [31] BS2000  
**Datenkommunikationssystem** \*  
Technische Beschreibung  
(Data Communication System, Technical Description)

*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.

*Contents*

Functions and principles of implementation of

- BCAM (Basic Communication Access Method)
- DCAM (Data Communication Access Method)
- TIAM (Terminal Interactive Access Method)
- RBAM (Remote Batch Access Method)
- VTSU-B (Virtual Terminal Support Basic).

*Order number*

U3214-J-Z53-1

- [32] BS2000  
**Programmiersystem** \*  
Technische Beschreibung  
(Programming System, Technical Description)

*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.

*Contents*

Functions and principles of implementation of

- the linkage editor
- the static loader
- the Dynamic Linking Loader
- the debugging aids
- the program library system.

*Order number*

U3216-J-Z53-1

- [33] BS2000  
**Systemadministration** \*  
Technische Beschreibung  
(System Management, Technical Description)

*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.

*Contents*

Functions and principles of implementation of

- static loading (UGEN - Universal GENERator, SIR - System Install and Restore)
- dynamic loading (startup, DLL - Dynamic Linking Lader, DSSM - Dynamic SubSystem Management)
- a monitoring system (SM2 - Software Monitor 2)
- software diagnosis (SLED - Self-Loading Emergency Dump routine, CDUMP, SODA, SODUMP, trace manager)
- hardware diagnosis (TDP - Test and Diagnosis Program)
- system termination (shutdown, crash).

*Order number*

U3217-J-Z53-1

The publications marked with an \* are available in German only.



# Contents

<b>Preface</b>	<b>1</b>
Changes in BS2000 V10.0A since the last version BS2000 V9.5A	2
Functions of the user commands	5
Command formats	7
Continuation lines	9
– Continuation mechanism for SSMCOPT=N	9
– Continuation mechanism for SSMCOPT=Y	10
Conventions used in command descriptions (metacharacters)	12
List of user commands	14
<b>List of user commands and brief description according to application areas</b>	<b>21</b>
Commands for controlling job execution	22
Commands for file processing	26
Commands for device and volume reservation	29
Commands for program control	30
Commands for interrogating current values	31
Commands for transaction mode (DCAM)	33
Commands for operation of terminals	33
SPOOL jobs	35
Commands for job variable functions	36
Commands for multiprocessor system users	37
Commands for remote file access (RFA)	37
Commands for file transfer (FT)	37
Commands for messages	38
<b>User commands</b>	<b>39</b>
Organization of the command descriptions	39
Rules for abbreviating command input	39
Default value	40
Job, task, process	41
Job class	42
Names	43
Path name	43
Fully-qualified file name	44
Wildcards in the catalog ID and file name	46
ABEND Terminate current job	47

ABORT	Abort procedure	49
ADD-ISAM-POOL-LINK	Define ISAM pool link name	52
APPLICATION	Enter DCAM application in CLT	53
AUDIT	Log branch instruction addresses	56
BCNTRL	Suppress operator messages	60
BREAK	Request system mode	62
CALL	Initiate procedure file	64
CANCEL	Cancel job	76
CANCEL-FILE-TRANSFER	Cancel file transfer	80
CATALOG	Process catalog entry	83
CATJV	Catalog job variable	100
CHANGE	Change TFT entry	106
CONNECTION	Enter virtual connection in CLT	108
COPY	Copy file	111
CREATE-ISAM-POOL	Create or join ISAM pool	120
DATA	Open spoolin file	123
DCLJV	Define JV link name	127
DELETE-ISAM-POOL	Delete ISAM pool/sever link to ISAM pool	128
DELON	Delete ON command	129
DO	Initiate procedure file	130
DROP	Cancel HOLD status	133
END	Close spoolin file	134
ENDON	Terminate ON statement sequence	135
ENDP	Terminate procedure file	136
ENTER	Process ENTER job	138
EOF	Indicate end-of-file for SYSDTA	156
ERAJV	Delete job variable	158
ERASE	Erase file	162
ESCAPE	Interrupt procedure	188
EXECUTE	Load and start program	194
FILE	Define file attributes / control file processing	209
	Function overview of the FILE command	214
FSTATUS	Request catalog information	265
GETJV	Output job variable value	303
GETUS	Read user switches	305
HELP	Display message text	306
HOLD	Place TFT entry in hold status	308
IMPORT	Create catalog entry for private files	310
INTR	Start interrupt routine for loaded program	322
LOAD	Load program	324
LOGOFF	Terminate job	327
LOGON	Initiate job	329
MODIFY-JOB	Modify job attributes	342
MODIFY-JV-CONDITIONALLY	Check and set job variable	348

MODIFY-MSG-ATTRIBUTES	Define language for message output	351
MRSSTA	Display multiprocessor system status	353
MSGCONTROL	Add or delete (task-specific) message files	354
ON	Initiate conditional execution of a command sequence	356
OPTION	Specify logging during job execution	363
PARAMETER	Control compilation/translation	368
PASSWORD	Specify password	369
PAUSE	Inform operator and wait	372
PRINT	Output (print) file	373
PRIORITY	Modify job or run priority	434
PROCEDURE	Define procedure file attributes	436
PSWORD	Protect user ID by means of password	441
PUNCH	Output file	444
RDTFT	Request information from TFT and TST	464
RELEASE	Delete TFT entry	473
REMARK	Insert remark into command file	475
REMOVE-ISAM-POOL-LINK		
	Remove ISAM pool link name from the table of pools	476
RESTART	Restart program from checkpoint	477
RESUME	Switch from system to program mode	483
RFAEND	Clear down RFA connection	485
RFASTART	Set up RFA connection	486
RJOB	Assign name to batch job	489
RLOGOFF	Terminate batch terminal input	495
RLOGON	Initiate batch terminal input	497
RMSG	Issue message from batch terminal	499
ROUT	Request job output	501
RSTART	Activate batch terminal	504
RSTATUS	Request status of remote batch jobs	506
RSTOP	Deactivate batch terminal	509
RTI	Return to interrupted procedure	510
SECURE	Request resources	513
SECURE-RESOURCE-ALLOCATION	Request resources	520
	Reserve (disk) files, disks and disk devices	523
	Reserve (tape) files, tapes and tape devices	526
SETJV	Set job variable	533
SETSW	Read or reset job switches	535
	Use of job switches in BS2000	537
SETUS	Reset user switches	544
SET-SS-OPTIONS	Release address space reserved for subsystems	546
SHOW-CJC-STATUS	Display information on CJC functions	547
SHOW-DEVICE-CONFIGURATION	Request configuration information	550
SHOW-DEVICE-STATUS		
	Request information on device assignment and monitoring	553

SHOW-DISK-DEFAULTS	
Interrogate default values for disk parameters	556
SHOW-DISK-STATUS Display allocation and disk parameters	557
SHOW-FILE Display file or library member on screen	560
SHOW-FILE-TRANSFER Display information about FT jobs	567
SHOW-JOB-CLASS Request information on job classes in use	571
SHOW-JV-LINK Display job variable entries	572
SHOW-MOUNT-PARAMETERS Display MOUNT settings	573
SHOW-MSG-DEFAULTS List message files	574
SHOW-RESOURCE-ALLOCATION	
Display allocations and outstanding operator actions	576
SHOW-SPOOL-CHARACTER-SETS Interrogate character sets	578
SHOW-SPOOL-DEVICES List printers	580
SHOW-SPOOL-FORMS List print forms	582
SHOW-SPOOL-PARAMETERS Show parameter list	584
SHOW-TAPE-STATUS Display tape allocation and monitoring	589
SHOW-USER-ATTRIBUTE Request information on user ID	592
SKIP Branch according to task switch	597
SKIPJV Conditional branch in command sequence	600
SKIPUS Branch according to user switch	602
SPARAM Compress SPOOL output	604
STAJV Output job variable attributes	605
STAM Read MRSCAT entries	607
STATUS Request information on jobs and system load	611
STEP Set procedure section	638
SYSFILE Assign system files	640
Input from SYSDTA or SYSIPT	644
Output to SYSLST or SYSOPT	646
Output to SYSLSTn	648
Output to SYSOUT	650
Output from SYSLST or SYSOPT to printer	651
Output from SYSOPT to floppy disk	652
Output for the Dynamic Linking Loader DLL	653
Input from SYSDTA to floppy disk	654
SYSTATUS Output information on system files	658
TCHNG Change terminal characteristics	660
TRANSFER-FILE Transfer file	666
TYPE Inform operator	683
VERIFY Reconstruct file	684
WAIT Initiate conditional wait state	690
WHEN Enter conditional wait state	692
WRITE-ACCOUNTING-RECORD Write user accounting record	695

**BS2000 system files** . . . . . **697**

**Appendix** . . . . . **705**

Device type code . . . . . 705

Meaning of output columns for the SHOW commands . . . . . 712

**Abbreviations** . . . . . **723**

**References** . . . . . **725**

# BS2000/OSD-BC V1.0

User Commands (ISP Format)



## 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](mailto:ts.fujitsu.com).

The Internet pages of Fujitsu Technology Solutions are available at

[http://ts.fujitsu.com/...](http://ts.fujitsu.com/)

and the user documentation at <http://manuals.ts.fujitsu.com>.

Copyright Fujitsu Technology Solutions, 2009

## 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](mailto:ts.fujitsu.com).

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter

[http://de.ts.fujitsu.com/...](http://de.ts.fujitsu.com/), und unter <http://manuals.ts.fujitsu.com> finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009